Rough set-based neuro-fuzzy system.

Ang, Kai Keng.

2008

Ang, K. K. (2008). Rough set-based neuro-fuzzy system. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/2487>

<https://doi.org/10.32657/10356/2487>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:37:17 SGT

Rough set-based Neuro-Fuzzy System

Ang Kai Keng

School of Computer Engineering

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

2008

Rough set-based Neuro-Fuzzy System

by

Ang Kai Keng

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy
May 2007

Abstract

Neuro-fuzzy systems is a popular hybridization in soft computing that abstracts a fuzzy model from given numerical examples using neural learning techniques to formulate accurate predictions on unseen samples. The fuzzy model incorporates the human-like style of fuzzy reasoning through a linguistic model that comprises if-then fuzzy rules and linguistic terms described by membership functions. However, modeling data using neuro-fuzzy systems involves the contradictory requirements of interpretability versus accuracy. Prevailing research that focused on accuracy employed optimization that resulted in membership functions that derailed from human-interpretable linguistic terms. In addition, the modeling of high-dimensional data requires a large number of if-then fuzzy rules that exceeds human level interpretation. This thesis focuses on increasing interpretability without compromising accuracy using a novel hybrid intelligent Rough set-based Neuro-Fuzzy System (RNFS), which synergizes rough set-based knowledge reduction with neuro-fuzzy systems. RNFS directly addresses the problems with the following contributions:

A Supervised Pseudo Self-Evolving Cerebellar (SPSEC) algorithm is proposed to generate human-interpretable membership functions in RNFS from numerical data. SPSEC employs a novel two-stage neural learning technique that first constructs a cerebellar-like structure in which neurons with high trophic factors evolves to form membership functions that relate intimately to the probability distributions of the data and concomitantly reconcile with defined semantic properties of linguistic variables.

A Rough Set-based Pseudo Outer-Product (RSPOP) algorithm is proposed to identify fewer if-then fuzzy rules that facilitate human interpretation in RNFS from numerical data. RSPOP integrates the sound theory of knowledge reduction in rough set theory with the Hebbian learning technique to identify non-redundant if-then fuzzy rules.

Existing neuro-fuzzy systems employ feature selection algorithms using the filter approach to model high-dimensional data. Prevailingly, the wrapper approach, which is considered to yield superior accuracy, is not employed. A Mutual Information-based Rough Set Reduction (MIRSR) algorithm is proposed to perform feature selection using the wrapper approach in RNFS. MIRSR synergizes the selection of features that result in high MI using SPSEC with the reduction of redundant features using RSPOP to yield features with high relevance and low redundancy.

Existing neuro-fuzzy systems are not capable of rejecting classifications with low confidence in modeling classification data. A Possibilistic Confidence Measure (PoCM) is proposed for estimating the classification confidence of RNFS. PoCM is useful for increasing the accuracy of RNFS by rejecting the classification of an unseen sample if the confidence is not sufficiently high.

The RNFS is used in a large collection of experiments to model real life data that involves predicting traffic flow, stock price and cancerous microarray gene expressions. Results demonstrated that the RNFS reduces computational complexity, improves the interpretability of the linguistic neuro-fuzzy system by generating membership functions that correspond to human-interpretable linguistic terms and identifying significantly fewer if-then fuzzy rules, as well as improves the accuracy of the prediction or classification.

Thesis Supervisor: Dr. Quek Hiok Chai

Title: Assoc. Prof. in Division of Computer Science, School of Computer Engineering

Acknowledgements

I would like to express my gratitude to my thesis advisor, Assoc. Prof. Quek Hiok Chai, for his patience, guidance and the freedom he had given me to explore my research throughout my Ph.D. candidature.

I would like to thank the Center for Computational Intelligence (C2I) and the lab technicians, Tan Swee Huat and Lau Boon Chee, for providing the facilities conducive for research. I would also like to thank my friends and colleagues in C2I for the fruitful discussions involving research as well as academic coursework; namely, Goh Ghee Ming, Quah Kian Hong, Guo Zaiyi, Ting Chan Wai and Tung Whye Loon. I am also grateful to Richard Jenson from the Department of Computer Science, University of Wales for providing me with the new Fuzzy-Rough Feature Selection (FRFS2) algorithm.

I would like to express my gratitude to my wife Li Hwa for her motivation and her patience in reading earlier drafts of this thesis. I would also like to express my gratitude to my parents for their continued support in my education.

Finally, I would like to express my appreciation to the School of Computer Engineering, Nanyang Technological University, and the Singapore Millennium Foundation for funding my scholarship.

Table of Contents

Abstract	i
Acknowledgements.....	iii
Table of Contents	iv
List of Figures	vii
List of Tables.....	xii
List of Abbreviations	xv
Chapter 1 Introduction	1
1.1 Background.....	1
1.2 Motivation.....	2
1.2.1 Neuro-fuzzy systems.....	2
1.2.2 Rough set theory	4
1.3 Organization of thesis	4
Chapter 2 Literature Review.....	5
2.1 Introduction.....	5
2.2 Neural Networks	5
2.2.1 Historical notes	7
2.2.2 Advantages and issues in modeling data	9
2.3 Fuzzy Systems	10
2.3.1 Historical notes	11
2.3.2 Advantages and issues in modeling data	13
2.3.3 Interpretability versus accuracy	14
2.4 Neuro-Fuzzy Systems	15
2.4.1 Generation of membership function	16
2.4.1.1 Clustering: Fuzzy C-Means (FCM) algorithm.....	23
2.4.1.2 Clustering: Learning Vector Quantization (LVQ) algorithm	24
2.4.1.3 Probability to possibility: Bijective transform	25
2.4.2 Identification of if-then fuzzy rules	27
2.4.2.1 MI-based feature selection using filter approach.....	31
2.4.2.2 MI-based Parzen Window (MIPW) algorithm	32
2.4.2.3 MI-based Best Individual Feature Parzen Window (MIBIFPW) algorithm.....	34
2.4.3 Reasoning methods.....	35
2.4.4 Pseudo Outer-Product based Fuzzy Neural Network	36
2.4.4.1 Learning process	41
2.4.4.2 Pseudo Outer-Product (POP) algorithm.....	42
2.4.5 Historical notes	44
2.4.6 Advantages and issues in modeling data	46
2.5 Rough Set Theory	47
2.5.1 Upper and lower approximation	47
2.5.2 Knowledge reduction.....	48
2.5.2.1 Knowledge representation system.....	49
2.5.2.2 Rule consistency	51
2.5.2.3 Attribute reduction.....	51
2.5.2.4 Decision rule reduction.....	51
2.5.3 Historical notes	52
2.5.4 Advantages and issue in analyzing data.....	52
2.6 Confidence in modeling classification data	54
2.6.1 Probabilistic rejection rule	54
2.6.2 Possibilistic rejection rule.....	55
2.6.3 Advantages and issues in modeling classification data	57
2.7 Summary.....	58

Chapter 3	Rough set-based Neuro-Fuzzy System	59
3.1	Introduction	59
3.2	Architecture	62
3.3	Supervised Learning	63
3.3.1	Generation of membership functions.....	64
3.3.1.1	Supervised Pseudo Self-Evolving Cerebellar (SPSEC) algorithm	64
3.3.2	Identification of if-then fuzzy rules	69
3.3.2.1	Rough Set-based Pseudo Outer-Product (RSPOP) algorithm	69
3.4	MI-based feature selection using wrapper approach.....	76
3.4.1	MI-based Rough Set Reduction (MIRSR) algorithm	78
3.5	Computation of confidence measure	80
3.5.1	Probabilistic Confidence Measure (PrCM).....	80
3.5.2	Possibilistic confidence measure (PoCM)	82
3.6	Summary	83
Chapter 4	Comparative Studies.....	85
4.1	Introduction.....	85
4.2	Generation of membership function	85
4.2.1	Experiment: UCI Sonar dataset	86
4.2.2	Discussion.....	91
4.3	Identification of if-then fuzzy rules	92
4.3.1	Experiment: Nakanishi datasets.....	93
4.3.1.1	Experiment 1: A nonlinear system	93
4.3.1.2	Experiment 2: Human operation of a chemical plant	97
4.3.1.3	Experiment 3: Daily price of a stock in a stock market	99
4.3.1.4	Discussion	101
4.3.2	Experiment: Traffic flow prediction	103
4.3.2.1	Discussion	107
4.4	Feature selection for modeling high dimensional data	109
4.4.1	Experiment: UCI Sonar dataset	109
4.4.2	Discussion.....	113
4.5	Estimating classification confidence.....	114
4.5.1	Experiment: Artificially generated dataset	114
4.5.2	Discussion.....	124
4.6	Summary	125
Chapter 5	Financial Case Study: Stock Trading.....	127
5.1	Introduction.....	127
5.2	Forecasting stock price	127
5.3	Experiment: Artificially generated stock price data	131
5.4	Stock trading systems	135
5.5	Experiments: Trading of stock from real world stock market	140
5.5.1	Neptune Orient Lines (NOL).....	140
5.5.2	NOL with different trading parameters	144
5.5.3	Development Bank of Singapore (DBS)	145
5.6	Summary	149
Chapter 6	Biomedical Case Study: Classifying cancerous samples	151
6.1	Introduction.....	151
6.2	Rejecting classification with low confidence.....	152
6.2.1	Experiment: Pima Indian diabetes dataset	152
6.3	Classifying high dimensional biomedical data	164
6.3.1	Experiment: Bhattacharjee lung cancer dataset	165
6.3.1.1	Feature selection and classification	165
6.3.1.2	Estimation of classification confidence and rejection.....	168
6.3.2	Experiment: Schummer ovarian cancer dataset.....	173

6.3.2.1 Feature selection and classification	173
6.3.2.2 Estimation of classification confidence and rejection.....	176
6.3.3 Experiment: Golub Leukemia dataset.....	180
6.3.3.1 Feature selection and classification	180
6.3.3.2 Estimation of classification confidence and rejection.....	182
6.3.4 Consolidated experimental results	188
6.3.4.1 Feature selection and classification	188
6.3.4.2 Estimation of classification confidence and rejection.....	190
6.4 Summary	191
Chapter 7 Conclusions and Future Work	193
7.1 Conclusions.....	193
7.2 Recommendations for future work	196
References	199
Author's Publications.....	212

List of Figures

Figure 2-1: Architecture of multilayered feedforward neural network trained using backpropagation learning	6
Figure 2-2: Taxonomy of neural network models adapted from (Sima & Orponen, 2003)	7
Figure 2-3: Architecture of a fuzzy system.....	10
Figure 2-4: Taxonomy of fuzzy system models.....	11
Figure 2-5: (a) Trapezoidal membership function $\mu_T(x; 3,4,6,8)$ and (b) Gaussian fuzzy membership functions $\mu_G(x; 0.5,4,1,6)$	17
Figure 2-6: An example of a linguistic variable, adapted from (Klir & Yuan, 1995).....	21
Figure 2-7: Architecture of POPFNN	36
Figure 2-8: POP in the identification of Output Label nodes for k^{th} rule node	41
Figure 3-1: Relationships between the proposed algorithms and the RNFS	61
Figure 3-2: Architecture of Rough set-based Neuro-Fuzzy System	62
Figure 3-3: SPSEC neural learning process (a) Steps 1-2 constructs a basic cerebellar structure consisting of regularly spaced neurons, (b) Steps 3-4 active neurons stabilize through the update of trophic factors while competitors die, (c) Steps 5 surviving neurons form the resulting membership functions that reconcile with semantics of linguistic variables	68
Figure 4-1: Inherent distributions of x_{52} from the training data of the UCI Sonar dataset (a) probability distribution $p(x_{52} \omega)$ estimated using Parzen Window and normal optimal smoothing; and (b) posterior probability $p(\omega x_{52})$ computed from $p(x_{52} \omega)$ using Bayes rule	87
Figure 4-2: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the FCM algorithm with parameters $c =$ (a) 3 clusters; and (b) 4 clusters	87
Figure 4-3: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the LVQ algorithm with parameter $c =$ (a) 3 clusters; and (b) 4 clusters	88
Figure 4-4: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the DIC algorithm with parameters $\{SLOPE, STEP\} =$ (a) $\{0.2, 0.2\}$; (b) $\{0.3, 0.2\}$; (c) $\{0.2, 0.4\}$; and (d) $\{0.3, 0.4\}$	89
Figure 4-5: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the Bijective transform from the inherent posterior probability distribution	90
Figure 4-6: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the proposed SPSEC algorithm.....	90
Figure 4-7: Five manually constructed membership functions for modeling a nonlinear system from Nakanishi datasets	94
Figure 4-8: Plot of the prediction results of the abstracted model using the proposed RSPOP and POP against actual test data of a nonlinear system from Nakanishi dataset	97
Figure 4-9: Plot of the prediction results of the abstracted model using the proposed RSPOP and POP against actual test data of a chemical plant from Nakanishi datasets.....	99

Figure 4-10: Plot of the prediction results of the abstracted model using the proposed RSPOP and POP against actual test data of the daily price of a stock in a stock market from Nakanishi datasets	100
Figure 4-11: Location of site 29 along the Pan Island Expressway in Singapore, where the traffic flow dataset (Tan, 1997) is collected.....	104
Figure 4-12: Plot of the traffic flow density of the three straight lanes along PIE at site 29 .	105
Figure 4-13: Recall and prediction of lane 1 traffic density for $\tau = 5$ using the abstracted model where the if-then fuzzy rules are identified using the proposed RSPOP (a) recall for CV1; (b) prediction of CV2&3; (c) recall for CV2; (d) prediction of CV1&3; (e) recall for CV3; (f) prediction of CV1&2	106
Figure 4-14: Consolidated experimental results on the accuracy of traffic flow prediction using the various abstracted models (a) average MSE of lane 1; (b) average R^2 of lane 1; (c) average MSE of lane 2; (d) average R^2 of lane 2; (e) average MSE of lane 3; (f) average R^2 of lane 3	107
Figure 4-15: Consolidated results on the classification accuracies performed using 3-fold cross-validations without selection bias on the UCI Sonar dataset.....	113
Figure 4-16: Plot of the generated dataset and support vectors where $\mu_1=[0,0]$ and $\mu_2=[1,1]$ with standard deviation of (a) $\sigma=0.2$ and (b) $\sigma=0.3$	116
Figure 4-17: Surface plot of the output score of Support Vector Machine trained using the artificially generated dataset with (a) $\sigma=0.2$ and (b) $\sigma=0.3$	116
Figure 4-18: Surface plot of the output score of Fisher Linear Discriminant trained using the artificially generated dataset with (a) $\sigma=0.2$ and (b) $\sigma=0.3$	117
Figure 4-19: Plot of the component probability distributions (a) $p(X_1 \omega)$, (b) $p(X_2 \omega)$, component posterior probability distribution (c) $p(\omega X_1)$, (d) $p(\omega X_2)$ of the artificially generated dataset with $\sigma=0.2$; plot of the component probability distribution (e) $p(X_1 \omega)$, (f) $p(X_2 \omega)$, component posterior probability distribution (g) $p(\omega X_1)$, and (h) $p(\omega X_2)$ of the artificially generated dataset with $\sigma=0.3$...	119
Figure 4-20: Plot of the joint posterior probability distribution $p(X \omega)$ of the artificially generated dataset with (a) $\sigma=0.2$ and (b) $\sigma=0.3$; and plot of the PrCM CM(X) of the artificially generated dataset with (c) $\sigma=0.2$ and (d) $\sigma=0.3$	120
Figure 4-21: Plot of the membership functions generated using the proposed SPSEC algorithm that characterize the possibility distribution (a) $\pi_{\omega X}(X_1, \omega)$ and (b) $\pi_{\omega X}(X_2, \omega)$ of the artificially generated dataset with $\sigma=0.2$; and plot of the membership functions generated using the proposed SPSEC algorithm that characterize the possibility distribution (c) $\pi_{\omega X}(X_1, \omega)$ and (d) $\pi_{\omega X}(X_2, \omega)$ of the artificially generated dataset with $\sigma=0.3$	121
Figure 4-22: Plot of the joint possibility distribution $\pi_{\omega X}(x, \omega_i)$ of the artificially generated dataset with (a) $\sigma=0.2$ and (b) $\sigma=0.3$; and plot of the PoCM CM(x) of the artificially generated dataset with (c) $\sigma=0.2$ and (d) $\sigma=0.3$	123
Figure 5-1: Time-delayed price forecast approach adapted from (Ang & Quek, 2006b)	129
Figure 5-2: Time-delayed price difference forecast approach adapted from (Ang & Quek, 2006b)	130
Figure 5-3: Artificially generated stock price data adapted from (Ang & Quek, 2006b)	132

Figure 5-4: Membership functions (a) manually constructed for inputs $\Delta y(T) $ to $\Delta y(T-9) $ in RNFS-1; and (b) manually constructed for output $\Delta y(T+1) $ in RNFS-1 and generated for inputs $\Delta y(T) $ to $\Delta y(T-9) $ and output $\Delta y(T+1) $ using SPSEC in RNFS-2	133
Figure 5-5: A trading decision model based on the technical analysis approach without forecast	136
Figure 5-6: A trading decision model based on the technical analysis approach with forecast adapted from (Moody et al., 1998).....	138
Figure 5-7: A forecast bottleneck free trading decision model with RNFS forecast adapted from (Ang & Quek, 2006b).....	139
Figure 5-8: Plot of price $y(T)$ and trading decisions $F(T)$ using the ST-RNFS and ST-basic models against time T on NOL	141
Figure 5-9: Plot of the portfolio end values $R(T)$ from the stock trading models against time T on NOL	141
Figure 5-10: Magnified results of Figure 5-8 from time $T=4500$ to $T=4580$	143
Figure 5-11: (a) Manually constructed input membership functions employed in RNFS; (b) manually constructed output membership functions employed in RNFS; and (c) if-then fuzzy rules identified using RSPOP employed in RNFS on NOL	144
Figure 5-12: Plot of the portfolio end values $R(T)$ from the stock trading models against time T on NOL using another set of trading parameters	144
Figure 5-13: Plot of price $y(T)$ and trading decisions $F(T)$ using the ST-RNFS and ST-basic models against time T on DBS.....	146
Figure 5-14: Plot of the portfolio end values $R(T)$ from the stock trading models against time T on DBS.....	146
Figure 5-15: (a) Manually constructed input membership functions employed in RNFS; (b) manually constructed output membership functions employed in RNFS; and (c) if-then fuzzy rules identified using RSPOP employed in RNFS on DBS.....	147
Figure 6-1: Plot of the joint posterior probability distribution (a) $p(X \omega_1)$ for class $\omega=1$; (b) $p(X \omega_2)$ for class $\omega=2$; (c) $p(X \omega_i)$ for both $\omega=1$ and 2; and (d) plot of the PrCM $CM(X)$ of NBPW against FLD transformed features t_1 and t_2 of the Pima Indians diabetes dataset	154
Figure 6-2: Plot of the membership functions generated using the proposed SPSEC employed in RNFS that characterize the possibility distribution (a) $\pi_{\omega X}(X_2, \omega)$; (b) $\pi_{\omega X}(X_6, \omega)$; and (c) $\pi_{\omega X}(X_8, \omega)$ of the Pima Indians diabetes dataset.....	155
Figure 6-3: Plot of the joint possibility distribution (a) $\pi_{\omega X}(X, \omega_1)$ for $\omega=1$, (b) $\pi_{\omega X}(X, \omega_2)$ for $\omega=2$, (c) $\pi_{\omega X}(X, \omega_i)$ for both $\omega=1$ and 2 and (d) plot of the PoCM $CM(X)$ of RNFS against transformed FLD features t_1 and t_2 of the Pima Indians diabetes dataset	156
Figure 6-4: Plot of the error rate e and rejection rate r against the PrCM of NBPW on the Pima Indians diabetes dataset	158
Figure 6-5: Plot of the error rate e and rejection rate r against the PrCM $CM(X)$ of NBPW on the test data of the Pima Indians diabetes dataset using a objective function of (a) $J=2e^2+r^2$; and (b) $J=4e^2+r^2$	159
Figure 6-6: Plot of the error rate e and rejection rate r against the PoCM $CM(X)$ of RNFS on the test data of the Pima Indians diabetes dataset using a objective function of (a) $J=2e^2+r^2$; and (b) $J=4e^2+r^2$	160

Figure 6-7: Plot of the error rate e and rejection rate r against the PrCM $CM(x)$ of NBPW on the 3-fold cross-validation sets of the Pima Indians diabetes dataset using a objective function of (a) $J=2e^2+r^2$; and (b) $J=4e^2+r^2$	162
Figure 6-8: Plot of the error rate e and rejection rate r against the PoCM $CM(X)$ of RNFS on the 3-fold cross-validation sets of the Pima Indians diabetes dataset using a objective function of (a) $J=2e^2+r^2$; and (b) $J=4e^2+r^2$	163
Figure 6-9: Consolidated results on the classification accuracies performed using (a) 3-fold; and (b) 10-fold cross-validations without selection bias and without rejection on the Bhattacharjee lung cancer dataset	167
Figure 6-10: Plot of the joint probability (a) $p(X \omega_1)$ for $\omega=1$, (b) $p(X \omega_2)$ $\omega=2$, (c) $p(X \omega_i)$ for both $\omega=1$ and 2 and (d) plot of the PrCM $CM(X)$ of NBPW against FLD transformed features t_1 and t_2 of one of the three cross-validation sets from the Bhattacharjee lung cancer dataset	169
Figure 6-11: Plot of the membership functions generated using the proposed SPSEC employed in RNFS that characterize the possibility distributions (a) $\pi_{\omega X}(X_{388}, \omega)$; (b) $\pi_{\omega X}(X_{393}, \omega)$; and (c) $\pi_{\omega X}(X_{270}, \omega)$ of one of the three cross-validation sets from the Bhattacharjee lung cancer dataset	169
Figure 6-12: Plot of the joint possibility distribution (a) $\pi_{\omega X}(X, \omega_1)$ for $\omega=1$, (b) $\pi_{\omega X}(X, \omega_2)$ for $\omega=2$, (c) $\pi_{\omega X}(X, \omega_i)$ for both $\omega=1$ and 2 and (d) plot of the PoCM $CM(X)$ of RNFS against transformed features t_1 and t_2 of one of the cross-validation sets from the Bhattacharjee lung cancer dataset.....	171
Figure 6-13: Plot of the error rate e and rejection rate r against (a) PrCM $CM(X)$ of NBPW and (b) PoCM $CM(X)$ of RNFS on the 3-fold cross-validation sets of the Bhattacharjee lung cancer dataset using a objective function of $J=4e^2+r^2$	172
Figure 6-14: Consolidated results on the classification accuracies performed using (a) 3-fold; and (b) 10-fold cross-validations without selection bias and without rejection on the Schummer ovarian cancer dataset	175
Figure 6-15: Plot of the joint posterior probability distribution (a) $p(X \omega_1)$ for $\omega=1$, (b) $p(X \omega_2)$ $\omega=2$, (c) $p(X \omega_i)$ for both $\omega=1$ and 2 and (d) plot of the PrCM $CM(X)$ against FLD transformed features t_1 and t_2 of one of the three cross-validation sets from the Schummer ovarian cancer dataset	177
Figure 6-16: Plot of the membership functions generated using the proposed SPSEC employed in RNFS that characterize the possibility distributions (a) $\pi_{\omega X}(X_{1483}, \omega)$, (b) $\pi_{\omega X}(X_9, \omega)$, (c) $\pi_{\omega X}(X_{93}, \omega)$; and (d) $\pi_{\omega X}(X_{1526}, \omega)$ of one of the three cross-validation sets from the Schummer ovarian cancer dataset	178
Figure 6-17: Plot of the error rate e and rejection rate r against the PrCM $CM(X)$ of NBPW on the 3-fold cross-validation sets of the Schummer ovarian cancer dataset using a objective function of $J=4e^2+r^2$	179
Figure 6-18: Consolidated results on the classification accuracies performed using (a) 3-fold; and (b) 10-fold cross-validations without selection bias and without rejection on the Golub Leukemia cancer dataset	182
Figure 6-19: Plot of the joint probability (a) $p(X \omega_1)$ for $\omega=1$, (b) $p(X \omega_2)$ $\omega=2$, (c) $p(X \omega_i)$ for both $\omega=1$ and 2 and (d) plot of the PrCM $CM(X)$ of NBPW against FLD transformed features t_1 and t_2 of one of the three cross-validation sets from the Golub Leukemia dataset.....	183

Figure 6-20: Plot of the membership functions generated using the proposed SPSEC employed in RNFS that characterize the possibility distributions (a) $\pi_{\omega \chi}(X_{6885}, \omega)$; (b) $\pi_{\omega \chi}(X_{1685}, \omega)$; (c) $\pi_{\omega \chi}(X_{758}, \omega)$; (d) $\pi_{\omega \chi}(X_{1114}, \omega)$; (e) $\pi_{\omega \chi}(X_{2354}, \omega)$; (f) $\pi_{\omega \chi}(X_{4680}, \omega)$; and (g) $\pi_{\omega \chi}(X_{2642}, \omega)$ of one of the three cross-validation sets from the Golub Leukemia dataset.....	184
Figure 6-21: Plot of the joint possibility distribution (a) $\pi_{\omega \chi}(X, \omega_1)$ for $\omega=1$, (b) $\pi_{\omega \chi}(X, \omega_2)$ for $\omega=2$, (c) $\pi_{\omega \chi}(X, \omega_i)$ for both $\omega=1$ and 2 and (d) plot of the PoCM $CM(X)$ of RNFS against transformed features t_1 and t_2 of t_1 and t_2 of one of the cross-validation sets from the Golub Leukemia dataset.....	186
Figure 6-22: Plot of the error rate e and rejection rate r against (a) PrCM $CM(X)$ of NBPW and (b) PoCM $CM(X)$ of RNFS on the 3-fold cross-validation sets of the Golub Leukemia dataset using a objective function of $J=4e^2+r^2$	187
Figure 6-23: Averaged results on the classification accuracies from the three datasets performed using 3-fold and 10-fold cross-validations without selection bias and without rejection	190
Figure 7-1: A consolidated view of the contributions of RNFS in this thesis	193

List of Tables

Table 2-1: List on some development of neural networks in chronological order	8
Table 2-2: List on some development of fuzzy systems in chronological order.....	12
Table 2-3: List on some development of neuro-fuzzy systems in chronological order	45
Table 2-4: An example of an information system adopted from (Quilan, 1983).....	48
Table 2-5: An example of the Knowledge Representation System (KRS) using attribute-value table adopted from (Pawlak, 1991)	50
Table 2-6: List on some development of rough set theory in chronological order	53
Table 4-1 Consolidated results on the generation of membership functions of x_{52} from the training data of the UCI Sonar dataset	91
Table 4-2: If-then fuzzy-rules identified using RSPOP for modeling a nonlinear system from Nakanishi datasets.....	95
Table 4-3: Semantic interpretation of the if-then fuzzy rules in Table 4-2.....	97
Table 4-4: If-then fuzzy rules identified using RSPOP for modeling the human operation of a chemical plant from Nakanishi datasets.....	98
Table 4-5: If-then fuzzy rules identified using RSPOP for modeling daily price of a stock in a stock market from Nakanishi datasets.....	100
Table 4-6: Consolidated experimental results on the accuracy of abstracted models on three Nakanishi datasets.....	102
Table 4-7: Consolidated benchmark of using POP against RSPOP in identifying if-then fuzzy rules on three Nakanishi datasets	102
Table 4-8: Consolidated benchmark on the average number of if-then fuzzy rules and average MSE of the various abstracted model on traffic flow prediction	108
Table 4-9: Consolidated experimental results of the classification accuracies of using various MI-based feature selection algorithms and classifiers performed using 3-fold cross-validations without selection bias on the UCI Sonar dataset.....	112
Table 5-1: Experimental results of time-delayed price difference forecast approach using the selected models on artificially generated stock price data	133
Table 5-2: Averaged experimental results of time-delayed price difference forecast approach using selected models on 20 sets of artificially generated stock price data	134
Table 6-1: Confusion matrix on the test data of the Pima Indians diabetes dataset using NBPW without rejection	152
Table 6-2: If-then fuzzy rules identified using the proposed RSPOP employed in RNFS on the Pima Indians diabetes dataset.....	155
Table 6-3: Confusion matrix on the test data of the Pima Indians diabetes dataset using RNFS without rejection	156
Table 6-4: Confusion matrix on the test data of the Pima Indians diabetes dataset using NBPW with optimal rejection using PrCM	160
Table 6-5: Confusion matrix on the test set of the Pima Indians diabetes dataset using RNFS with optimal rejection using PoCM	161
Table 6-6: Confusion matrix on the cross-validation sets of the Pima Indians diabetes dataset using NBPW with optimal rejection using PrCM.....	162

Table 6-7: Confusion matrix on the cross-validation sets of the Pima Indians diabetes dataset using RNFS with optimal rejection using PoCM.....	163
Table 6-8: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the Pima Indians diabetes dataset	163
Table 6-9: Consolidated experimental results of the classification accuracies of using a selection of classifiers and a selection of feature selection algorithms performed using 3-fold and 10-fold cross-validations without selection bias and without rejection on the Bhattacharjee lung cancer dataset	166
Table 6-10: Confusion matrix on the 3-fold validation sets of the Bhattacharjee lung cancer dataset using NBPW without rejection	168
Table 6-11: If-then fuzzy rules identified using the proposed RSPOP employed in RNFS on one of the cross-validation sets from the Bhattacharjee lung cancer dataset	170
Table 6-12: Confusion matrix on the 3-fold validation sets of the Bhattacharjee lung cancer dataset using RNFS without rejection.....	170
Table 6-13: Confusion matrix on the cross-validation sets of the Bhattacharjee lung cancer dataset using NBPW and RNFS with optimal rejection using PrCM and PoCM respectively	172
Table 6-14: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the Bhattacharjee lung cancer dataset.....	172
Table 6-15: Consolidated experimental results of the classification accuracies of using a selection of feature selection algorithms and a selection of classifiers performed using 3-fold and 10-fold cross validations without selection bias and without rejection on the Schummer ovarian cancer dataset	174
Table 6-16: Confusion matrix on the 3-fold validation sets of the Schummer ovarian cancer dataset using NBPW without rejection	176
Table 6-17: If-then fuzzy rules identified using the proposed RSPOP employed in RNFS on one of the cross-validation sets from the Schummer ovarian cancer dataset....	178
Table 6-18: Confusion matrix on the validation sets of the Schummer ovarian cancer dataset using RNFS without rejection.....	179
Table 6-19: Confusion matrix on the cross-validation sets of the Schummer ovarian cancer dataset using the NBPW with reject using PrCM	179
Table 6-20: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM on the 3-fold cross-validation sets of the Schummer ovarian cancer dataset.....	180
Table 6-21: Consolidated experimental results of the classification accuracies of using a selection of feature selection algorithms and a selection of classifiers performed using 3-fold and 10-fold cross validations without selection bias and without rejection on the Golub Leukemia dataset.....	181
Table 6-22: Confusion matrix on the 3-fold validation sets of the Golub Leukemia dataset using NBPW without rejection	183
Table 6-23: If-then fuzzy rules identified using the proposed RSPOP employed in RNFS on one of the cross-validation sets from the Golub Leukemia dataset.....	185
Table 6-24: Confusion matrix on the 3-fold validation sets of the Golub Leukemia dataset using NBPW without rejection	185

Table 6-25: Confusion matrix on the cross-validation sets of the Golub Leukemia dataset using NBPW and RNFS with optimal rejection using PrCM and PoCM respectively	187
Table 6-26: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the Golub Leukemia dataset	187
Table 6-27: Consolidated experimental results of the classification accuracies from the three datasets using a selection of feature selection algorithms and a selection of classifiers performed using 3-fold and 10-fold cross validations without selection bias and without rejection	189
Table 6-28: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the three datasets.....	190
Table 7-1: A summary of research achievements in this thesis	193

List of Abbreviations

AARS	Approximate Analogical Reasoning Schema
ANFIS	Adaptive-Network-based Fuzzy Inference System
CART	Classification And Regression Tree
CM	Confidence Measure
CMAC	Cerebellar Model Articulation Controller
CRI	Compositional Rule of Inference
CV	Cross-Validation
DBS	Development Bank of Singapore
DENFIS	Dynamic Evolving Neural-Fuzzy Inference System
DIC	Discrete Incremental Clustering
EFuNN	Evolving Fuzzy Neural Network
EMA	Exponential Moving Average
FCM	Fuzzy C-Means
FFNN-BP	Feed-Forward Neural Network trained using Back Propagation
FKP	Fuzzy Kohonen Partitioning
FLD	Fisher Linear Discriminant
k-NN	k nearest neighbor
LVQ	Learning Vector Quantization
MI	Mutual Information
MINBPW	Mutual Information Naïve Bayesian Parzen Window
MIPW	Mutual Information Parzen Window
MIRSR	Mutual Information Rough Set Reduction
MSE	Mean Square Error
NBPW	Naïve Bayesian Parzen Window
NOL	Neptune Orient Lines
PoCM	Possibilistic Confidence Measure
POP	Pseudo Outer-Product
POPFNN	Pseudo Outer-Product-based Fuzzy Neural Network
PrCM	Probabilistic Confidence Measure
RBFN	Radial Basis Function Network
RNFS	Rough set-based Neuro-Fuzzy System
RSPOP	Rough Set-based Pseudo Outer-Product
SPSEC	Supervised Pseudo Self-Evolving Cerebellar
ST-basic	Stock trading without forecast
ST-DENFIS	Stock trading with DENFIS forecast
ST-ideal	Stock trading with ideal forecast
ST-RNFS	Stock trading with RNFS forecast
SVM	Support Vector Machine
TSK	Takagi-Sugeno-Kang
TVR	Truth Value Restriction

Chapter 1 Introduction

1.1 Background

This thesis is about soft computing, or more generally, computational intelligence. Prevailing, the definition of Computational intelligence is a confusing, hotly debated issue with no consensus in sight (Duch, 2007). Nevertheless, it is defined as subset of Artificial Intelligence (Bezdek, 1994) that extends the endeavor of understanding how we humans think to the building of intelligent entities (Poole, Mackworth, & Goebel, 1998). In the research field of computational intelligence, soft computing plays a key role in the conception and design of hybrid intelligent systems.

Soft computing, coined by Professor Lotfi Zadeh (1994), differs from the conventional hard computing in that it is tolerant of imprecision, uncertainty and partial truth. It is this inherent tolerance for imprecision and uncertainty in humans that allows us to perform daily tasks such as understanding distorted speech, deciphering sloppy handwriting, comprehending the nuances of natural language, or driving a vehicle in dense traffic. Soft computing uses the human mind as a role model and aims to formalize the cognitive processes we humans employ so effectively in performing daily tasks (Zadeh, 1994). Soft computing encompasses a collection of computational methodologies in the field of computational intelligence and machine learning that mainly comprises of neural network, fuzzy logic and evolutionary algorithms (Dubois & Prade, 1998), as well as other computational methodologies such as probabilistic networks, rough set theory, chaos theory and parts of learning theory (Jang, Sun, & Mizutani Eiji, 1997).

1.2 Motivation

The core problem in soft computing is about bridging the gap between subjective

knowledge and objective data (Dubois & Prade, 1998). There are two approaches of addressing this problem; namely, modeling data in which a function is built to mimic the data accurately, and abstracting data in which a system is built to produce articulated knowledge preferably in natural language form (Dubois & Prade, 1998). The emphasis of the former is on the ability to reproduce what has been observed. Neural networks with their prominent learning capabilities inspired from biological systems are highly suitable in this approach. On the other hand, the emphasis of the latter is on the ability to explain the data in a human interpretable way. Fuzzy systems with the ability of modeling linguistic terms that are expressions of human language are likewise highly effective in this approach. In fuzzy systems, linguistic expressions are formulated from explicit knowledge in the form of if-then fuzzy rules where the linguistic terms of the antecedents and consequents are fuzzy sets. However, the parameters of these linguistic expressions are sometimes difficult to specify and have to be manually tuned (Nauck, Klawonn, & Kruse, 1997). In contrast, although neural networks are capable of learning from data, they are black box models (Sjoberg et al., 1995) and thus soliciting knowledge from neural networks is not a straightforward task. Hence, a neural network is capable of modeling data, but it is difficult for a user to learn from it. On the other hand, a user can learn from a fuzzy system easily, but it is difficult to model data without explicit knowledge from a user.

1.2.1 Neuro-fuzzy systems

Neuro-fuzzy hybridization is a methodology in soft computing (Mitra & Hayashi, 2000) known as Fuzzy Neural Networks or Neuro-Fuzzy Systems in the literature (Buckley & Hayashi, 1994; Buckley & Yoichi Hayashi, 1995; Lin & Lee, 1996; Nauck et al., 1997; Mitra & Hayashi, 2000).

Neuro-fuzzy systems are empowered with the ability to abstract a fuzzy model from given numerical examples using neural learning techniques in which the abstracted fuzzy model is able to formulate accurate predictions on new data. The abstracted fuzzy model comprises linguistic expressions in the form of if-then fuzzy rules and linguistic terms described by membership functions. Thus, the main strength of neuro-fuzzy systems is that they are universal approximators (Feuring & Lippe, 1999; Wang & Wei, 2000; Ying, Ding, Li, & Shao, 1999; Tikk, Kóczy, & Gedeon, 2003) with the ability to solicit humanly interpretable if-then rules (Guillaume, 2001). Hence, a neuro-fuzzy system learns and at the same time, a user can learn from it as well.

However, the use of neuro-fuzzy systems in abstracting data involves two contradictory requirements in fuzzy modeling: interpretability versus accuracy (Casillas, Cordon, Herrera, & Magdalena, 2003). In practice, only one of the two properties prevails. Thus the fuzzy modeling research field is divided into: *linguistic fuzzy modeling* that is focused on interpretability, mainly using the Mamdani model (Mamdani & Assilian, 1975); and *precise fuzzy modeling* that is focused on accuracy, mainly using the Takagi-Sugeno-Kang model (Takagi & Sugeno, 1985; Sugeno & Kang, 1988). Prevailing research has focused on increasing the accuracy of the model while neglecting the interpretability of the final model (Casillas et al., 2003). Some examples include unconstrained optimization that resulted in membership functions that do not represent human-interpretable linguistic terms (de Oliveira, 1999b), or the use of a large number of if-then fuzzy rules where any semantic interpretability is lost (Tikk & Baranyi, 2003; Nauck, 2003). Moreover, if the number of if-then rules is bounded as a practical limitation, then the universal approximation property is lost (Moser, 1999). Therefore, abstracting a fuzzy model that is not humanly interpretable utterly derails the fundamental purpose of using neuro-fuzzy systems.

1.2.2 Rough set theory

Recently, rough set theory (Pawlak, 1991), one of the methodologies in soft computing, has shown to provide efficient techniques of finding hidden patterns in data (Pawlak, 2002). Rough set-based methods have shown the potential for feasible feature selection with the ability to significantly reduce the pattern dimensionality in neural networks (Swinarski & Skowron, 2003; Shang & Shen, 2002). This motivates the investigation of synergizing rough set-based methods with neuro-fuzzy systems and focusing on the objective of increasing the interpretability of the abstracted model without compromising the accuracy.

1.3 Organization of thesis

This thesis consists of seven chapters. Chapter 2 presents a literature review of the fields related to this research work, which include neural networks, fuzzy systems, neuro-fuzzy systems and rough sets. Chapter 3 presents the proposed Rough set-based Neuro-Fuzzy System (RNFS), which comprises of the Rough Set-based Pseudo Outer-Product (RSPOP) algorithm for identifying if-then fuzzy rules, the Supervised Pseudo Self-Evolving Cerebellar (SPSEC) algorithm for generating membership functions, the Mutual Information-based Rough Set Reduction (MIRSR) algorithm for performing feature selection on high-dimensional data, and the Possibilistic Confidence Measure (PoCM) for estimating the classification confidence. Chapter 4 presents comparative studies of using the aforementioned algorithms employed in the RNFS. Chapters 5-6 present extensive experimental results of using the RNFS on real life data that involves predicting stock price and high-dimensional cancerous microarray gene expressions respectively. Finally, Chapter 7 concludes this thesis and provides directions for future work.

Chapter 2 Literature Review

2.1 Introduction

This chapter presents a brief review of the components in soft computing that are relevant to this thesis, specifically, neural networks, fuzzy systems and rough set theory. The advantages and issues on modeling data using neural networks and fuzzy systems are highlighted, and how existing neuro-fuzzy systems overcome these issues are discussed. The Pseudo Outer-Product Fuzzy Neural Network (POPFNN) (Zhou & Quek, 1996; 1999; Ang, Quek, & Pasquier, 2003), which is used as the foundation for the development of the proposed neuro-fuzzy system, is reviewed in more detail. Prevailing issues on using neuro-fuzzy systems and rough set theory in modeling data are also discussed.

2.2 Neural Networks

Neural networks are motivated by the recognition that the human brain computes in an entirely different way from the conventional digital computer (Haykin, 1999). They are generally constructed to resemble the human brain in the ability to acquire knowledge through a learning process and the ability to store the acquired knowledge in terms of synaptic weights. A neural network is a parallel, distributed processing network made up of simple computational units called neurons. The neurons that serve as inputs to the network are called input neurons, those that serve as outputs to the network are called output neurons while the remaining ones are called hidden neurons. These neurons are generally arranged in layers and connected in an oriented graph manner that represents the architecture of the network (Sima & Orponen, 2003). Figure 2-1 illustrates an example on the architecture of the multilayered feedforward perceptron neural network model trained using the backpropagation learning method (Rumelhart, Hinton, & Williams, 1986).

Figure 2-1: Architecture of multilayered feedforward neural network trained using backpropagation learning

Prevailingly, neural network models are classified according to the type of neurons used such as perceptron, radial basis function, winner-take-all or spiking neurons; the computation model such as deterministic, probabilistic or noisy; the architecture such as feedforward or recurrent (Sima & Orponen, 2003). The taxonomy of neural network models is illustrated in Figure 2-2.

The difference between recurrent and feedforward architecture is that the former augments the latter with feedbacks from the output units to the input units. The difference between probabilistic and deterministic computation is that the former augments the latter with additional random input units. Noisy computation also augments the deterministic computation with the notion of noise in spiking neurons in order to simulate imprecise firing times.

Figure 2-2: Taxonomy of neural network models adapted from (Sima & Orponen, 2003)

2.2.1 Historical notes

Table 2-2 lists some of the research development in neural networks in chronological order. For more comprehensive reviews, please refer to (Zurada, 1992; Haykin, 1999; Sima & Orponen, 2003).

Table 2-1: List on some development of neural networks in chronological order

1943	Warren McCulloch and Walter Pitts (1943) pioneered the work in neural networks and artificial intelligence (Russell & Norvig, 2003). They formalized a model of <i>artificial neurons</i> in which each neuron is characterized by either on or off and showed that a network of such connected neurons is able to compute any computable function.
1949	Donald Olding Hebb (1949) postulated that a learning process intensifies the connection between two neurons if they are active at the same time. Hebb's learning rule, now called <i>Hebbian learning</i> , is commonly accepted as the fundamental neurobiological principle of learning (Müller, Reinhardt, & Strickland, 1995; Nauck et al., 1997) and remains influential even in modern neural network literature (Russell & Norvig, 2003).
1958	Frank Rosenblatt (1958) introduced the novel <i>supervised learning</i> method in his work on the <i>perceptron</i> .
1960	Bernie Widrow and Marcian Hoff Jr. (1960) introduced the least mean-square algorithm and used it to formulate the Adaptive Linear Element (<i>Adaline</i>). The difference between the perceptron and the Adaline lies in the training procedure.
1972	Stephen Grossberg (1972; 1976a; 1976b; 1980), established <i>unsupervised learning</i> in a principle of self-organization known as <i>adaptive resonance theory</i> (ART).
1975	James Albus (1975a; 1975b) introduced the <i>Cerebellar Model Articulation Controller</i> (CMAC) based on his understanding of the function of the cerebellum.
1982	Teuvo Kohonen (1982; 1989) proposed a network consists of <i>winner-take-all</i> neurons in self-organizing feature maps. John Hopfield (1982) introduced new class of neural network with feedbacks known as <i>recurrent networks</i> .
1983	Andrew Barto, Richard Sutton and Chuck Anderson (1983) introduced <i>reinforcement learning</i> , which is capable of learning in the absence of a teaching signal.
1986	David Rumelhart, Geoffrey Hinton and Ronald Williams (1986) introduced the <i>backpropagation algorithm</i> , a popular supervised learning algorithm for the training multi-layer feedforward neural networks. A review is available in (Widrow & Lehr, 1990).
1988	David Broomhead and David Lowe (1988) designed a new feed-forward neural network using <i>radial basis functions</i> .
1995	Corinna Cortes and Vladimir Vapnik (1995) introduced <i>support vector machines</i> which employed supervised learning for solving classification, regression and density estimation problems.
1999	Wolfgang Maass and Christopher Bishop (1999) introduced networks of <i>spiking neurons</i> as alternatives to the perceptron, since they are more biologically plausible than artificial neurons.

2.2.2 Advantages and issues in modeling data

The advantages of modeling data using neural networks are:

1. They are able to learn from numerical data without explicit specification of the functional or distribution form of the underlying model (Haykin, 1999).
2. They are universal function approximators that can approximate any function with arbitrary accuracy (Hornik, Stinchcombe, & White, 1989; Hornik, 1991).
3. They comprise interconnections of nonlinear neurons and thus are nonlinear models that are flexible in modeling complex real world data (Roadknight, Balls, Mills, & Palmer-Brown, 1997).
4. They are able to estimate the posterior probabilities of the numerical data, which provides the basis for performing classification and statistical analysis (Richard & Lippmann, 1991).

The issues on modeling data using neural networks are:

1. They are black box models (Sjoberg et al., 1995) in which the modeling is performed entirely from available numerical data using no additional a priori knowledge by considering a sufficiently general structure (Söderström & Stoica, 1989). Thus, they are unable to explain the embedded knowledge in a human interpretable way and how a particular decision is arrived at (Mitra & Hayashi, 2000).
2. Since they are black box models, they are also unable to incorporate a priori knowledge and thus have to acquire knowledge from scratch.
3. Their users have to deal with a large number of potentially necessary parameters (Sjoberg et al., 1995). Some examples of these parameters are choice of neural network model, choice of radial basis function, number of neurons and number of hidden layers.

2.3 Fuzzy Systems

Fuzzy systems are popular soft computing framework based on the concepts of fuzzy set theory, if-then fuzzy rules and fuzzy reasoning (Jang et al., 1997). *Fuzzy systems* (Dubois & Prade, 1980; Cox, 1999; Nedjah & Mourelle, 2005; Chen & Pham, 2006) due to their multidisciplinary nature, are also know by other names such as *fuzzy inference systems* (Jang et al., 1997), *fuzzy expert systems* (Kandel, 1992), *fuzzy logic systems* (Mendel, 2001), *fuzzy rule-based systems* (Lotfi, Andersen, & Tsoi, 1996; Cordon & Herrera, 1999), *fuzzy models* (Sugeno & Kang, 1988; Takagi & Sugeno, 1985), *fuzzy associative memories* (Kosko, 1992) and *fuzzy logic controllers* (Lee, 1990a; 1990b). Figure 2-3 shows the architecture of a fuzzy system, which consists of the following five fundamental components (Jang et al., 1997; Mendel, 2001):

- *Dictionary* – defines the linguistic terms used by the if-then fuzzy rules in the form of fuzzy sets that are described mathematically by membership functions.
- *Rule base* – contains the linguistic expressions in the form of if-then fuzzy rules.
- *Reasoning mechanism* – contains the *inference engine* that performs the inference procedure specified by the *reasoning method* using the if-then fuzzy rules, membership functions and the fuzzy inputs to derive a fuzzy output.
- *Fuzzification* – converts the crisp inputs into appropriate fuzzy inputs.
- *Defuzzification* – converts the fuzzy output into crisp output.

Figure 2-3: Architecture of a fuzzy system

Prevailing, fuzzy systems are classified by the differences in the consequents of the if-then fuzzy rules (Jang et al., 1997). The taxonomy of fuzzy system models is illustrated in Figure 2-4.

Figure 2-4: Taxonomy of fuzzy system models

Section 2.3.3 discusses the Mamdani (Mamdani & Assilian, 1975) and Takagi-Sugeno-Kang fuzzy system models (Takagi & Sugeno, 1985) in more detail. The Tsukamoto fuzzy system model (Tsukamoto, 1979) is similar to the Mamdani fuzzy system model, but uses fuzzy sets with monotonic membership functions to represent the consequent of each fuzzy if-then rule. In addition, the overall output of the Tsukamoto fuzzy system model is a weighted average of the crisp value from each if-then fuzzy rule. Hence, it is not commonly used because it is not as interpretable as the Mamdani or the Takagi-Sugeno-Kang fuzzy system models (Jang et al., 1997).

2.3.1 Historical notes

Table 2-2 lists some of the research development in fuzzy systems in chronological order. For a review on fuzzy sets and fuzzy logic, please refer to (Klir & Yuan, 1995). For a compiled selection of papers from Professor Lotfi Zadeh, please refer to (Yager, Ovchinnikov, Tong, & Nguyen, 1987). For more comprehensive reviews, please refer to these recent handbooks on fuzzy sets theory and its applications (Dubois & Prade, 2000; Höhle & Rodabaugh, 1998; Nguyen & Sugeno, 1998; Bezdek, Dubois, & Prade, 1999; Zimmermann, 1999; Slowinski, 1998; Bezdek, Keller, Krisnapuram, & Pal, 1999).

Table 2-2: List on some development of fuzzy systems in chronological order

1965	Lotfi Zadeh (1965) pioneered the work in fuzzy systems. He initiated the theory of <i>fuzzy sets</i> and created an important point in the evolution of the modern concept of uncertainty.
1972	Sheldon Chang and Lotfi Zadeh (1972) introduced the idea of applying fuzzy sets in control problems.
1973	Lotfi Zadeh (1973; 1976) introduced an approach of describing the behavior of complex systems using <i>linguistic variables</i> and fuzzy algorithm. This work laid the foundation for fuzzy logic control and most of the current applications of <i>fuzzy logic</i> . Other topics discussed include the extension principle, fuzzy sets of type 2, marginal and conditional restrictions, separability and non-interaction, and the <i>compositional rule of inference</i> .
1976	Ebrahim Mamdani (1976; Mamdani & Assilian, 1975) synthesized the first fuzzy controller that comprised of a linguistic model of 24 if-then fuzzy rules from the heuristic control rules stated by a human operator. This fuzzy system model is commonly known as the <i>Mamdani model</i> .
1978	Lotfi Zadeh (1978a; 1978b) introduced possibility theory and also developed Possibilistic Relational Universal Fuzzy (<i>PRUF</i>), a meaning representation language for natural language.
1979	Lotfi Zadeh (1979) contributed most of the ideas associated with <i>approximate reasoning</i> . His theory provided a powerful framework for reasoning of imprecise and uncertain information through the representation of propositions using fuzzy sets. Tsukamoto (1979) presented a fuzzy model similar to the Mamdani model but using fuzzy sets with monotonic membership functions to represent the consequent of each fuzzy if-then rule. This fuzzy system model is commonly known as the <i>Tsukamoto model</i> .
1985	Tomohiro Takagi and Michio Sugeno (1985; Sugeno & Kang, 1988) introduced a two-part qualitative model based on fuzzy logic: fuzzy modeling and linguistic approximation. This fuzzy system model is commonly known as the Sugeno model or the <i>Takagi-Sugeno-Kang model</i> .
1994	Lotfi Zadeh (1994) coined the term <i>soft computing</i> , which incorporates the field of neural networks, fuzzy systems and probabilistic reasoning.
1996	Lotfi Zadeh (1996) introduced the term <i>computing with words</i> based on the notion that human computes with words instead of numbers.
1997	Lotfi Zadeh (1997) introduced the <i>theory of fuzzy information granulation</i> , which is inspired by the ways humans granulate information and reason with it.
▼ 2006	Lotfi Zadeh (2006) presented a <i>generalized theory of uncertainty</i> where information is equated to a generalized constraint.

2.3.2 Advantages and issues in modeling data

The advantages of modeling data using fuzzy systems are:

1. They are able to express the behavior of the complex system with subjective knowledge in the absence of objective knowledge (Mendel, 2001). The subjective knowledge is represented by linguistic expressions such as *pressure is weak* that are usually impossible to quantify using precise mathematics while the objective knowledge is represented by mathematical first-principle models such as *Newton's laws of mechanics* that are usually used in science and engineering.
2. They are able to directly incorporate a priori knowledge from human experts into the model that comprises of linguistic expressions formulated in the form of if-then fuzzy rules (Jang et al., 1997).
3. They are universal approximators that can approximate any continuous function with arbitrary accuracy (Castro, 1995; Kreinovich, Nguyen, & Yeung, 2000).
4. They are non-linear models capable of modeling the parameters of an actual system with nonlinear dynamics (Takagi & Sugeno, 1985; Tay & Tan, 1997).

The issues of modeling data using fuzzy systems are:

1. They are unable to learn the membership functions and the if-then fuzzy rules in the knowledge base from available numerical data (Lin & Lee, 1996).
2. Their users have to deal with the optimization of the membership functions and the if-then fuzzy rules in the knowledge base from numerical data (Lin & Lee, 1996), which is a tedious and difficult process.
3. They are unable to model high dimensional data with high degree of interpretability since the number of if-then fuzzy rules required increases exponentially with the increase in number of inputs (Wang, 1998).

2.3.3 Interpretability versus accuracy

Modeling data using fuzzy systems is mainly characterized by two features that represent the quality of the abstracted fuzzy model (Casillas et al., 2003): interpretability and accuracy. Interpretability refers to the capability of the fuzzy model to express the behavior of the modeled system in an understandable way. It is a subjective factor, which depends on several factors such as the model structure, the number of input variables, the number of if-then fuzzy rules, the number of linguistic terms and the shape of the fuzzy sets. Accuracy refers to the capability of the fuzzy model in representing the modeled system. It is an objective factor, in which the closer the approximation of the fuzzy model is to the response of the real system that is being modeled, the higher the accuracy.

Professor Lotfi Zadeh (1973) stated in the *Principle of Incompatibility* that *as the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive*. Therefore, to achieve high degree of interpretability and accuracy is a contradictory endeavor and, in practice, only one of the two properties prevails (Casillas et al., 2003). Hence, the fuzzy modeling field is divided into two areas: *linguistic fuzzy modeling* that is focused on interpretability, mainly using the Mamdani model (Mamdani & Assilian, 1975) given in equation (2.1); and *precise fuzzy modeling* that is focused on accuracy, mainly using the Takagi-Sugeno-Kang (TSK) model (Takagi & Sugeno, 1985; Sugeno & Kang, 1988) given in equations (2.2) and (2.3) (Guillaume, 2001; Casillas et al., 2003).

$$R_k : \text{IF } x_1 \text{ is } A_{k,1} \text{ AND } \cdots \text{AND } x_{n_1} \text{ is } A_{k,n_1} \text{ THEN } y \text{ is } B_k. \quad (2.1)$$

$$R_k : \text{IF } x_1 \text{ is } A_{k,1} \text{ AND } \cdots \text{AND } x_{n_1} \text{ is } A_{k,n_1} \text{ THEN } y_k = \mathbf{a}_k \mathbf{x} + b_k, \quad (2.2)$$

$$y = \sum_{k=1}^{n_3} w_k(x) y_k, \quad (2.3)$$

where $\mathbf{x}, y$ are the input vector $x=[x_1, \dots, x_{n_1}]$ and output value respectively; $A_{k,i}, B_k$ are the linguistic terms with fuzzy sets associated defining their meaning; n_1 is the number of inputs; and n_3 is the number of if-then fuzzy rules. The consequents in equation (2.2) are linear functions of the inputs that appear garbled for human interpretation whereas the consequents in equation (2.1) are simply linguistic terms. Therefore, the TSK model has decreased interpretability but increased representative power compared against the Mamdani model. For a more comprehensive coverage on interpretability versus accuracy, please refer to (Casillas et al., 2003).

2.4 Neuro-Fuzzy Systems

Neural networks and fuzzy systems are very popular techniques in soft computing (Zadeh, 1994) capable of universal approximation of nonlinear numerical data. However, they are complementary in the design of computationally intelligent systems for modeling data (Lin & Lee, 1996). Each has its own advantages and issues. Neural networks but not fuzzy systems are capable of learning from numerical data. On the other hand, fuzzy systems but not neural networks are capable of expressing interpretable human-like subject knowledge. Neuro-fuzzy hybridization synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Thus, neuro-fuzzy systems are capable of abstracting a fuzzy model from given numerical examples using neural learning techniques. The abstracted model incorporates the human-like style of fuzzy reasoning through a linguistic model that comprises of if-then fuzzy rules and linguistic terms described by membership functions.

The architecture of neuro-fuzzy systems generally resembles the multilayered neural-network architecture in Figure 2-1 as well as the fuzzy systems architecture in Figure 2-3. Figure 2-7 in section 2.4.4 illustrates an example on the dual neural-network-like and fuzzy-system-like resemblance using the architecture of the Pseudo Outer-Product Fuzzy Neural Network. A neuro-fuzzy system thus consists of three conceptual components in fuzzy systems (Jang & Sun, 1995; Mendel, 2001): *dictionary*, *rule base* and *reasoning mechanism*. Hence, generating membership functions for the dictionary, identifying if-then fuzzy rules for the rule base, and specifying the reasoning method for the reasoning mechanism are important aspects of neuro-fuzzy systems. The following sections describe these conceptual components with the emphasis on the interpretability of the abstracted model.

2.4.1 Generation of membership function

The generation of membership function is an important step in many applications of fuzzy theory (Medasani, Kim, & Krishnapuram, 1998). Most common membership functions are triangular, trapezoidal, Gaussian and bell-shaped (Mendel, 2001). Figure 2-5(a) and (b) depict trapezoidal and Gaussian membership functions, mathematically described by equations (2.4) and (2.5) respectively. Triangular and bell-shaped membership functions can be described by equation (2.4) using parameters such that $\beta=\gamma$ and by equation (2.5) using parameters such that $\alpha=\gamma$ and $\beta=\delta$, respectively.

$$\mu_T(x; \alpha, \beta, \gamma, \delta) = \left\{ \begin{array}{ll} 0 & x \leq \alpha \text{ or } x \geq \delta \\ \frac{x-\alpha}{\beta-\alpha} & \alpha < x < \beta \\ 1 & \beta \leq x \leq \gamma \\ \frac{\delta-x}{\delta-\gamma} & \gamma < x < \delta \end{array} \right\}, \quad (2.4)$$

$$\mu_G(x; \alpha, \beta, \gamma, \delta) = \begin{cases} e^{\frac{-(x-\beta)^2}{2\alpha^2}} & x < \beta \\ 1 & \beta \leq x \leq \delta \\ e^{\frac{-(x-\delta)^2}{2\gamma^2}} & x > \delta \end{cases}, \quad (2.5)$$

where $\alpha, \beta, \gamma, \delta$ are parameters of the membership function.

Figure 2-5: (a) Trapezoidal membership function $\mu_T(x; 3,4,6,8)$ and (b) Gaussian fuzzy membership functions $\mu_G(x; 0.5,4,1,6)$

Although fuzzy membership functions can be mathematically described using equations (2.4) and (2.5), there is no uniformity in the interpretation of what a fuzzy membership grade means. Three main semantics for membership functions exist in the literature; namely, *similarity*, *preference* and *uncertainty* (Dubois & Prade, 1997). Considering the degree of membership $\mu_F(u)$ of an element u in a fuzzy set F defined on the universe U , the three interpretations of this degree are:

- *Similarity* – is the proximity of u to a prototype element F (Bellman, Kalaba, & Zadeh, 1966). If u is exactly F then $\mu_F(u)$ is 1.
- *Preference* – is the preference in favor of selecting u where F represents a set of preferred objects (Bellman & Zadeh, 1970).
- *Uncertainty* – is the plausibility that a parameter x has value u , given that x is F . This is proposed by Zadeh (1978b) when he introduced possibility theory and approximate reasoning.

In addition, there are no measures available to evaluate the goodness or correctness of the membership functions generated (Medasani et al., 1998). As a result, there are several approaches to the generation of fuzzy membership functions:

- *Heuristics* – uses predefined membership functions where associated parameters are provided by human experts. This approach yields reasonably smooth membership functions that are easily manipulated by fuzzy operators, but the parameters have to be manually optimized (Medasani et al., 1998).
- *Histograms* – provides information for estimating the probability distributions of the data, which can be represented by parameterized functions such as Gaussian, thus directly yielding membership functions. This approach is easy to implement and the membership functions can be used for classifying data (Medasani et al., 1998), but the histograms of different classes frequently overlap, hence limiting the applicability for finding linguistic terms (Duch, Setiono, & Zurada, 2004). Moreover, the randomness represented by probability theory and the vagueness represented by fuzzy set theory are inherently different concepts (Kosko, 1992). Hence, the validation on the interpretation of membership functions generated using this method remains in question (Medasani et al., 1998).
- *Nearest neighbors* – assigns class memberships to a sample instead of a particular class, where the class memberships depend on the sample's distance from its k nearest neighbors (Keller, Gray, & Givens, 1985). This approach is the simplest, but generates membership functions that are not smooth (Medasani et al., 1998).
- *Feedforward neural networks* – employs neurons using sigmoid activation functions that are equivalent to fuzzy membership functions that can be easily extracted (Medasani et al., 1998; Duch, 2005). This approach is capable of generating complex membership functions for classifying data, but the

membership values are not necessarily indicative of the similarity of a feature to a class and are unpredictable in regions where there is no training data (Medasani et al., 1998).

- *Clustering* – partitions a dataset into subsets called clusters so that data in each subset are similar. This approach employs unsupervised self-organization which generates membership functions that efficiently covers the regions where numerical data are present (Lin, 1995), but the number of clusters must be known (Medasani et al., 1998). Although clustering algorithms such as the *Robust Agglomerative Gaussian Mixture Decomposition* (RAGMD) (Medasani et al., 1998), *Adaptive Resonance Theory* (ART) (Lin & Lin, 1997; Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992) and *Discrete Incremental Clustering* (Tung & Quek, 2002a) do not require the specification of the number of clusters, other parameters that affect the number of clusters generated are required. The parameters required are, namely, the retention ratio P in RAGMD (Medasani et al., 1998); the vigilance criterion ρ in ART (Lin & Lin, 1997); and the SLOPE and STEP in DIC (Tung & Quek, 2002a).
- *Probability to possibility transformations* – employs the possibility/probability consistency principle defined by Professor Lotfi Zadeh (1978b) to convert probability distribution function to possibility distribution function. This principle expresses a weak connection between possibility and probability where a fuzzy variable is associated with a possibility distribution represented by a membership in the same manner as a random variable is associated with a probability distribution. This approach assumes that the general shapes of the probability functions and possibility functions are similar and thus facilitates the generation of membership functions directly from normalized histograms (Medasani et al.,

1998). Therefore, the validation on the interpretation of membership functions generated using this method, which is similar to the histogram method, remains in question (Medasani et al., 1998).

Due to the number of different approaches available in the literature, it is difficult to choose a single approach of generating membership functions that works for most applications (Medasani et al., 1998). However, the clustering method is popularly employed in neuro-fuzzy systems to generate membership functions (Tung & Quek, 2002b; Panella & Gallo, 2005; Nedjah & Mourelle, 2005). For example, the *Learning Vector Quantization* algorithm (Kohonen, 1989) is popularly employed in Mamdani models (Lin, 1995; Zhou & Quek, 1996), and the *Fuzzy C-Means algorithm* (Bezdek, 1981) is popularly employed in TSK models (Flores-Sintas, Cadenas, & Martin, 1999; Li, Mukaidono, & Turksen, 2002; Panella & Gallo, 2005). Descriptions of the Fuzzy C-Means and Learning Vector Quantization algorithms are provided in Sections 2.4.1.1 and 2.4.1.2 respectively. A description on *Bijjective transform* (Dubois & Prade, 1983), which is based on the probability to possibility transform approach that has also gained interest in recent year (Masson & Denoeux, 2006), is provided in section 2.4.1.3. For a review on membership generation techniques, please refer to (Medasani et al., 1998).

Since the main advantage of using neuro-fuzzy systems is to abstract humanly interpretable linguistic expressions from available numerical data, the membership functions generated have to reconcile with the semantic properties of a linguistic variable (Casillas et al., 2003). A linguistic variable is formally defined by Zadeh (1975) with a quintuple $(L, T(L), U, G, M)$ where L is the name of the variable; $T(L)$ is the linguistic term set of L ; U is a universe of discourse; G is a syntactic rule that

generates $T(L)$; and M is a semantic rule that associates each $T(L)$ with its meaning. Each linguistic term is characterized by a fuzzy set that is described mathematically using a membership function.

Figure 2-6: An example of a linguistic variable, adapted from (Klir & Yuan, 1995)

Figure 2-6 illustrates an example of a linguistic variable x named $L = \text{“performance”}$ (Klir & Yuan, 1995). It is characterized by five linguistic terms $T(L) = \{\text{“very small”}, \text{“small”}, \text{“medium”}, \text{“large”}, \text{“very large”}\}$ where each of these linguistic terms is assigned one of the five trapezoidal membership functions by a semantic rule M . These trapezoidal membership functions cover the entire universe of discourse $U = [0, 100]$ of the base variable x . All the fuzzy sets described by the trapezoidal membership functions that characterize the linguistic terms of $T(L)$ are *normalized* and *convex*. In addition, all the linguistic terms followed a *partial ordering* such that $\text{“very small”} \prec \text{“small”} \prec \text{“medium”} \prec \text{“large”} \prec \text{“very large”}$. Although the definition of interpretability and its criteria are subjected to controversial discussion, interpretable linguistic variables are often associated with the shape and mutual overlapping of the membership functions (Mikut, Jakel, & Groll, 2005) as illustrated in Figure 2-6. Nevertheless, formal definitions on the semantic properties of interpretable linguistic variables were proposed (Mikut et al., 2005; de Oliveira,

1999a):

- *Coverage* – membership functions $\mu_{X_j}(x)$ that characterize linguistic terms where

$X_j \in T(L)$ cover the entire universe of discourse if $\forall x \in U, \exists X_j \in T(L)$ such that

$$\mu_{X_j}(x) > 0. \quad (2.6)$$

- *Normalized* – membership function $\mu_{X_j}(x)$ that characterizes a linguistic term

where $X_j \in T(L)$ is normalized if $\exists x \in X_j$ such that

$$\mu_{X_j}(x) = 1. \quad (2.7)$$

- *Convex* – membership function $\mu_{X_j}(x)$ that characterizes a linguistic term where

$X_j \in T(L)$ is convex if $\forall x, y, z \in X_j$

$$x \leq y \leq z \Rightarrow \mu_{X_j}(y) \geq \min(\mu_{X_j}(x), \mu_{X_j}(z)). \quad (2.8)$$

- *Ordered* – membership function $\mu_{X_j}(x)$ that characterize linguistic terms where

$X_j \in T(L) = \{X_1, X_2, \dots, X_j, \dots, X_n\}, j \in [1, n]$ is ordered if

$$X_1 \prec X_2 \prec \dots \prec X_j \prec \dots \prec X_n, \quad (2.9)$$

where the symbol $\prec$ denotes a partial ordering (Koczy & Hirota, 1993) such that

$X_1 \prec X_2$ denotes X_1 precedes X_2 . X_1 precedes X_2 if

$$\forall \alpha \in (0, 1]: \left(\inf \{ {}^\alpha X_1 \} < \inf \{ {}^\alpha X_2 \} \right) \wedge \left(\sup \{ {}^\alpha X_1 \} < \sup \{ {}^\alpha X_2 \} \right), \quad (2.10)$$

where ${}^\alpha X_j$ denotes an α -cut of a fuzzy set X_j ; and the α -cut of a fuzzy set X_j is the

crisp set ${}^\alpha X_j = \{x | X_j(x) \geq \alpha\}$.

The prevailing issue of generating membership function in neuro-fuzzy systems in modeling data is that existing methods focused on the optimization of membership functions to achieve high accuracy of the abstracted model, but resulted in membership functions that deviate from semantic properties of human-interpretable linguistic terms (de Oliveira, 1999b).

2.4.1.1 Clustering: Fuzzy C-Means (FCM) algorithm

- Step 1: Given data set $\tilde{\mathbf{X}} = \{\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_k, \dots, \tilde{X}_n\}$, define c as the number of clusters, m as the exponent weight and a small positive number ε as the terminating criterion.
- Step 2: Initialize iteration $T=0$ and randomly initialize fuzzy pseudo-partition $\mathbf{P}^{(0)}$. A fuzzy pseudo-partition of $\mathbf{P}$ is a family of fuzzy subsets $\mathbf{P} = \{P_1, P_2, \dots, P_c\}$ which satisfies equations (2.11) and (2.12).

$$\sum_{i=1}^c \mu_i(\tilde{X}_k) = 1 \quad \forall k \in \{1, 2, \dots, n\}, \quad (2.11)$$

$$0 < \sum_{k=1}^c \mu_k(\tilde{X}_i) < n \quad \forall i \in \{1, 2, \dots, c\}, \quad (2.12)$$

where $\mu_i(\tilde{X}_k)$ denotes the membership of $\tilde{X}_k$ in fuzzy subset P_i .

- Step 3: Compute the cluster centers $\tilde{V}_1^{(T)}, \tilde{V}_2^{(T)}, \dots, \tilde{V}_j^{(T)}, \dots, \tilde{V}_c^{(T)}$ for $\mathbf{P}^{(T)}$ using equation

$$\tilde{V}_j^{(T)} = \frac{\sum_{k=1}^n (\mu_j(\tilde{X}_k))^m \tilde{X}_k}{\sum_{k=1}^n (\mu_j(\tilde{X}_k))^m} \quad \text{for } j=1..c \quad (2.13)$$

- Step 4: Update $\mathbf{P}^{(T+1)}$ with equation

$$\mu_i^{(T+1)}(\tilde{X}_k) = \left(\sum_{j=1}^c \left(\frac{\|\tilde{X}_k - \tilde{V}_i^{(T)}\|^2}{\|\tilde{X}_k - \tilde{V}_j^{(T)}\|^2} \right)^{\frac{1}{m-1}} \right)^{-1} \quad \text{for } i=1..c, k=1..n. \quad (2.14)$$

If $\|\tilde{X}_k - \tilde{V}_i^{(T)}\|^2 = 0$, then $\mu_i^{(T+1)}(\tilde{X}_k) = 1$ and $\mu_j^{(T+1)}(\tilde{X}_k) = 0$ for $j=1..c, j \neq i$.

- Step 5: Compare $\mathbf{P}^{(T+1)}$ with $\mathbf{P}^{(T)}$ using equation

$$E = \|\mathbf{P}^{(T+1)} - \mathbf{P}^{(T)}\| = \sum_{j=1}^c \sum_{k=1}^n \left| \mu_j^{(T+1)}(\tilde{X}_k) - \mu_j^{(T)}(\tilde{X}_k) \right|. \quad (2.15)$$

If $E > \varepsilon$ then $T=T+1$ and go to step 3. If $E \leq \varepsilon$ then stop.

2.4.1.2 Clustering: Learning Vector Quantization (LVQ) algorithm

- Step 1: Given data set $\tilde{\mathbf{X}} = \{\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_k, \dots, \tilde{X}_n\}$, define c as the number of clusters, α as the learning constant where $0 < \alpha < 1$, a small positive ε as the terminating criterion and $T_{\max}$ as the maximum number of iterations.
- Step 2: Initialize iteration $T=0$, weights $\tilde{V} = \{\tilde{V}_1^{(0)}, \tilde{V}_2^{(0)}, \dots, \tilde{V}_j^{(0)}, \dots, \tilde{V}_c^{(0)}\}$ and initial learning constant $\alpha^{(0)} = \alpha$.
- Step 3: For $T = 0..T_{\max}$:

For $k = 1..n$:

- Find winner w using equation

$$\|\tilde{X}_k - \tilde{V}_w^{(T)}\| = \min_j (\|\tilde{X}_k - \tilde{V}_j^{(T)}\|) \text{ for } j = 1..c \quad (2.16)$$

- Update the weights of the winner with equation

$$\tilde{V}_w^{(T+1)} = \tilde{V}_w^{(T)} + \alpha^{(T)} (\tilde{X}_k - \tilde{V}_w^{(T)}) . \quad (2.17)$$

End for k

- Compute $E^{(T+1)}$ using equation

$$E^{(T+1)} = \|\tilde{V}^{(T+1)} - \tilde{V}^{(T)}\|^2 = \sum_{j=1}^c \|\tilde{V}_j^{(T+1)} - \tilde{V}_j^{(T)}\|^2 . \quad (2.18)$$

- If $E^{(T+1)} \leq \varepsilon$ stop, else adjust learning rate $\alpha^{(T+1)}$ to satisfy equations

$$\sum_{T=0}^{\infty} \alpha^{(T)} = \infty, \text{ and} \quad (2.19)$$

$$\sum_{T=0}^{\infty} (\alpha^{(T)})^2 < \infty . \quad (2.20)$$

End for T

One choice for the sequence of learning constants that satisfies equations (2.19) and (2.20) is given in equation

$$\alpha^{(T+1)} = \lambda \alpha^{(T)}, \quad (2.21)$$

where λ is the learning rate, $0 < \lambda \leq 1$.

2.4.1.3 Probability to possibility: Bijective transform

The kernel density estimator, also referred to as the Parzen Window estimator (Parzen, 1962), is a popular method for estimating the conditional probability distribution because of its excellent performance and firm theoretical foundation (Babich & Camps, 1996). Given the universe of discourse $\mathbf{U}$ of a d -dimensional variable $\mathbf{x}=[x_1, x_2, \dots, x_i, \dots, x_d]$, the conditional probability distribution of an attribute x_i that covers the universe of discourse U_i of the i^{th} dimension, can be estimated using the Parzen Window in equation

$$\hat{p}(\hat{x}_{i,p} | \omega) = \frac{1}{n_\omega} \sum_{k \in I_\omega} \phi(\hat{x}_{i,p} - x_{i,k}, h), \forall \hat{x}_{i,p} \in U_i, \quad (2.22)$$

where $x_{i,k}$ are the data samples for the i^{th} attribute; I_ω is the set of indices of the data samples belonging to class ω ; ϕ is a smoothing kernel function with a smoothing parameter h ; and $\hat{\mathbf{x}}_i = \{\hat{x}_{i,1}, \hat{x}_{i,2}, \dots, \hat{x}_{i,p}, \dots, \hat{x}_{i,n_p}\}$ are n_p evenly distributed points that cover U_i . Although various selections of ϕ and h are provided in (Parzen, 1962), the rectangular and Gaussian kernel functions are commonly used (Kwak & Chong-Ho Choi, 2002). The univariate Gaussian kernel is given by equation

$$\phi(x, h) = \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x^2}{2h^2}\right)}, \quad (2.23)$$

and the multivariate Gaussian kernel is given by equation

$$\phi(\mathbf{x}, h) = \frac{1}{(2\pi)^{d/2} h^d |\Sigma|^{1/2}} e^{-\left(\frac{\mathbf{x}^T \Sigma^{-1} \mathbf{x}}{2h^2}\right)}, \quad (2.24)$$

where Σ is the covariance matrix of d -dimensional vector of random variable $\mathbf{x}$.

While the use of the Parzen window addressed the problem of estimating $p(x|\omega)$, this estimation is still dependent on the choice of h . The *normal optimal smoothing strategy* in (Bowman & Azzalini, 1997), which offers an effective strategy for an

optimal choice of h that required very little computation, is given by equation

$$h^{opt} = \left(\frac{4}{3n} \right)^{\frac{1}{5}} \sigma, \quad (2.25)$$

where σ denotes the standard deviation of the distribution.

The posterior probability $p(\omega | \hat{x}_{i,p})$ of a data sample $\hat{x}_{i,p}$ can be computed using

Bayes rule in equation (2.26) with the data sample $\mathbf{x} = \hat{x}_{i,p}$

$$p(\omega | \mathbf{x}) = \frac{p(\mathbf{x} | \omega) P(\omega)}{p(\mathbf{x})}, \quad (2.26)$$

where $p(\omega | \mathbf{x})$ is the conditional probability of class ω given the data sample $\mathbf{x}$; $p(\mathbf{x} | \omega)$ is the conditional probability of $\mathbf{x}$ given class ω ; $P(\omega)$ is the prior probability of class ω , $\omega \in \{1, \dots, N_\omega\}$; N_ω is the total number of classes; and $p(\mathbf{x})$ is

$$p(\mathbf{x}) = \sum_{\omega=1}^{N_\omega} p(\mathbf{x} | \omega) P(\omega). \quad (2.27)$$

Various methods of transforming probability distribution to possibility distribution have been proposed in the literature (Masson & Denoeux, 2006; Klir, 1990; Civanlar & Trussell, 1986; Dubois & Prade, 1983). If the posterior probability of $\hat{x}_{i,p}$ is ordered such that $\hat{p}(\omega | \hat{x}_{i,1}) \geq \hat{p}(\omega | \hat{x}_{i,2}) \geq \dots \geq \hat{p}(\omega | \hat{x}_{i,p}) \geq \dots \geq \hat{p}(\omega | \hat{x}_{i,n_p})$, then a bijective transformation between probability and possibility (Dubois & Prade, 1983; Dubois & Prade, 1988) is defined as

$$\pi_{\omega|\mathcal{X}}(\hat{x}_{i,p}, \omega) = \sum_{j=1}^{n_p} \min(p(\omega | \hat{x}_{i,p}), p(\omega | \hat{x}_{i,j})), \quad (2.28)$$

where the possibility $\pi_{\omega|\mathcal{X}}(\hat{x}_{i,1}, \omega) = 1 \geq \pi_{\omega|\mathcal{X}}(\hat{x}_{i,2}, \omega) \geq \dots \geq \pi_{\omega|\mathcal{X}}(\hat{x}_{i,n_p}, \omega) \geq 0$.

2.4.2 Identification of if-then fuzzy rules

The main advantage of using neuro-fuzzy systems is to abstract humanly interpretable if-then fuzzy rules from available numerical data. Thus, the identification of interpretable if-then fuzzy rules forms the most important aspect in the design of neuro-fuzzy systems. The general form of the if-then fuzzy rules used in linguistic neuro-fuzzy systems based on the Mamdani model is given in equation (2.1), where the antecedents and consequent are linguistic terms. A variant of this form used in precise neuro-fuzzy systems based on the TSK model is given in equation (2.2), where the antecedents are linguistic terms but the consequent is a function of the input.

An example R_1 of an if-then fuzzy rule from a linguistic neuro-fuzzy system in the form given in equation (2.1), which describes the relationship between the temperature and pressure of a boiler to the heating power, is

R_1 : IF *temperature* is *very high* AND *pressure* is *low* THEN *power* is *very low* (2.29)

where *temperature* and *pressure* are the input linguistic variables x_1 and x_2 ; *power* is the output linguistic variable y ; the vague terms *very high*, *low* and *very low* are the linguistic terms $A_{1,1}$, $A_{1,2}$ and B_1 respectively. These linguistic terms are associated with fuzzy sets mathematically described by membership functions on the universe of discourse of *temperature*, *pressure* and *heat* values.

A number of research work on the identification of if-then fuzzy rules from numerical data has been reported (Lin & Lee, 1991; Hayashi, Nomura, Yamasaki, & Wakami, 1992; Ishibuchi, Tanaka, & Okada, 1994; Yager, 1994; Shann & Fu, 1995; Lozowski, Cholewo, & Zurada, 1996; Quek & Zhou, 2001; Tung & Quek, 2002b). They can be categorized into the following approaches (Quek & Zhou, 2001):

- *Expert knowledge* – employs information from human experts to identify linguistic terms and if-then fuzzy rules. Neural network learning techniques are then employed using available numerical data to perform optimization on the linguistic terms and if-then fuzzy rules. This approach is capable of fast learning convergence, but it is subjected to the availability of subjective information from the human experts (Quek & Zhou, 2001).
- *Supervised learning* – employs supervised learning that uses backpropagation to identify the if-then fuzzy rules. This approach is capable of modeling nonlinear data accurately (Ishibuchi et al., 1994; Lee, Lee, & Park, 1994), but results in a black box model without any semantic interpretability (Quek & Zhou, 2001).
- *Hybrid learning* – employs a first stage of unsupervised learning that comprises self-organized learning or clustering to generate the membership functions, and competitive learning to identify the if-then fuzzy rules; followed by a final stage of supervised learning that uses backpropagation to optimize the parameters of the input and output membership functions (Quek & Zhou, 2001; Nedjah & Mourelle, 2005). Although the unconstrained optimization in the final stage increases the accuracy of the abstracted model, it results in membership functions that are deviated from human-interpretable linguistic terms (de Oliveira, 1999b).

The hybrid learning approach is popularly employed in linguistic neuro-fuzzy systems using the Mamdani model such as *Fuzzy Adaptive Learning Control Network* (FALCON) (Lin & Lee, 1991), *Pseudo Outer-Product Fuzzy Neural Network* (POPFNN) (Zhou & Quek, 1996) and *Generic Self-Organizing Fuzzy Neural Network* (GenSoFNN) (Tung & Quek, 2002b). The Pseudo Outer-Product algorithm (Quek & Zhou, 2001), which is used to identify if-then fuzzy rules in the first stage of hybrid learning in POPFNN, is described in Section 2.4.4.2. For a review on the

identification of if-then fuzzy rules, please refer to (Mitra & Hayashi, 2000).

Generally, a large number of if-then fuzzy rules are needed in neuro-fuzzy systems to yield good approximation accuracy (Ying, 1994; Zeng & Singh, 1996; Ying & Chen, 1997; Ying et al., 1999). In addition, an increased number of if-then fuzzy rules is needed in the linguistic neuro-fuzzy systems to yield the same representative power of precise neuro-fuzzy systems (Ying et al., 1999). However, it is noted that if the number of if-then fuzzy rules is large in any of these models, then any semantic interpretability is lost (Tikk & Baranyi, 2003; Nauck, 2003). The prevailing issue of identifying if-then fuzzy rules in neuro-fuzzy systems in modeling data is that a large number of if-then fuzzy rules, which exceed human interpretation, are needed to model high dimensional data. A number of research work addressing the issue on the interpretability of precise neuro-fuzzy systems has been reported (Johansen & Babuška, 2003; Jin, 2000; Yen, Wang, & Gillespie, 1998). Existing linguistic neuro-fuzzy systems rely on the following approaches to improve interpretability:

- *Heuristic thresholds* – are used to identify fewer if-then fuzzy rules or perform rule pruning. For example, the LazyPOP algorithm employed in POPFNN uses three user-defined thresholds; namely, Noise Criterion (NC), Ambiguity Criterion (AC) and Focus Criterion (FC) in the identification and pruning of if-then fuzzy rules (Quek & Zhou, 2001). Similarly, the RULEMAP algorithm employed in GenSoFNN uses two user-defined thresholds; namely, ThresISP and ThresOSP in the identification of if-then fuzzy rules (Tung & Quek, 2002b). The use of heuristic thresholds reduces the number of if-then fuzzy rules identified, but the accuracy and the interpretability of the abstracted model are usually heavily dependent upon the selection of these heuristic threshold values. Moreover, if the number of if-then

fuzzy rules is bounded as a practical limitation, then the universal approximation property of the models does not hold anymore (Moser, 1999).

- *Feature selection* – involves selecting a subset of input variables called features to be used in the model (Casillas et al., 2003). There are mainly two approaches in the literature (Han & Kamber, 2001; Kohavi & John, 1997): *filter approach* where the features are selected independently from the modeling system and *wrapper approach* where features are selected using the modeling system. The filter approach employs statistics computed from the empirical data distribution (Chow & Huang, 2005) or semantics-preserving information contained within the empirical dataset (Shen & Chouchoulas, 2002; Jensen & Shen, 2004). The wrapper approach is considered to yield better performance at the expense of increased computational effort (Kohavi & John, 1997; Inza, Sierra, Blanco, & Larrañaga, 2002; Zhou & Mao, 2005). Existing linguistic neuro-fuzzy systems employ the filter approach to select a smaller subset of features so that the number of linguistic variables and if-then fuzzy rules identified on the smaller dataset are significantly reduced from the original dataset. For example, the *sparse logistic regression* is used in (Tan, Quek, & Ng, 2005) to select 9 out of 1536 features from the microarray gene expression dataset so that FALCON is able to model the smaller feature selected dataset. Similarly, the *Monte Carlo evaluative selection* (Quah & Quek, 2007) is used in (Tung & Quek, 2005) to select a maximum of 11 out of 12 558 features from the microarray gene expression dataset so that GenSoFNN is able to model the smaller feature selected dataset. Currently, the wrapper approach is not employed by existing linguistic neuro-fuzzy systems.

Prevailingly, Mutual Information (MI), which is a higher order statistics (Cover & Thomas, 2006), is considered to be a good indicator of relevance between variables.

Hence efforts to adopt MI in feature selection problems resulted in a series of research (Peng, Long, & Ding, 2005; Kwak & Chong-Ho Choi, 2002; Chow & Huang, 2005; Battiti, 1994). MI-based feature selection using filter approach is reviewed in section 2.4.2.1 in more detail.

2.4.2.1 MI-based feature selection using filter approach

The MI between random variables $\mathbf{X}$ and $\mathbf{Y}$ (Cover & Thomas, 2006) is given by equation

$$I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y} | \mathbf{X}), \quad (2.30)$$

where the entropy of the random variable $\mathbf{X}$ is given by equation

$$H(\mathbf{X}) = -\sum_{x \in \mathbf{X}} p(x) \log_2 p(x), \quad (2.31)$$

where $p(x)$ is the probability density function of x ; and the conditional entropy of $\mathbf{X}$ and $\mathbf{Y}$ is

$$H(\mathbf{Y} | \mathbf{X}) = -\sum_{x \in \mathbf{X}} \sum_{y \in \mathbf{Y}} p(x, y) \log_2 p(y | x). \quad (2.32)$$

In modeling classification data, the input features are usually continuous variables and the class has discrete values. Thus the MI between input features $\mathbf{X}$ and the class Ω is

$$I(\mathbf{X}; \Omega) = H(\Omega) - H(\Omega | \mathbf{X}) \quad (2.33)$$

where $\omega \in \Omega = \{1, \dots, N_\omega\}$ is a class from a total of N_ω ; and the conditional entropy is

$$H(\Omega | \mathbf{X}) = -\int_{\mathbf{X}} \sum_{\omega=1}^{N_\omega} p(\omega, x) \log_2 p(\omega | x) dx. \quad (2.34)$$

In the MI-based feature selection, the problem is defined as given an initial set F with d features, find the subset $S \subset F$ with k features that maximizes $I(S; \Omega)$ (Battiti, 1994).

In the filter approach, the conditional entropy $H(\Omega | \mathbf{X})$ in equation (2.34) is hard to compute since it is not easy to estimate $p(\omega | x)$. Recently, Kwak and Choi (2002) presented a method of estimating $p(\omega | x)$ and subsequently $H(\Omega | \mathbf{X})$ using Parzen

Window (Parzen, 1962). Using Bayes rule, $p(\omega|x)$ can be written as equation

$$p(\omega|x) = \frac{p(x|\omega)p(\omega)}{p(x)}, \quad (2.35)$$

and an estimate of $p(x|\omega)$ using Parzen Window (Parzen, 1962) is

$$\hat{p}(x|\omega) = \frac{1}{n_\omega} \sum_{i \in I_\omega} \phi(x - x_i, h), \quad (2.36)$$

where $\omega=1, \dots, N_\omega$; n_ω is the number of data samples belonging to class ω ; I_ω is the set of indices of the data samples belonging to class ω ; and ϕ is a smoothing kernel function with a smoothing parameter h in which the *normal optimal smoothing strategy* given in equation (2.25) provides an optimal choice.

Performing feature selection on a large number of features is computationally prohibitive. Therefore, suboptimal but computationally efficient methods such as *Sequential Forward Selection* (Whitney, 1971) and *Best Individual Features* are often used instead (Jain, Duin, & Mao, 2000). The Sequential Forward Selection method, also known as the greedy selection scheme, is adopted by Battiti (1994) in the *Mutual Information-based on Parzen Window* (MIPW) algorithm (Kwak & Chong-Ho Choi, 2002) to be described in section 2.4.2.2. The Best Individual Features method (Jain et al., 2000), which is a simpler method that evaluates all the features individually and selects only the best k features, is adopted in the *MI-based Best Individual Feature Parzen Window* (MIBIFPW) algorithm to be described in section 2.4.2.3.

2.4.2.2 MI-based Parzen Window (MIPW) algorithm

- Step 1: *Initialization*

Initialize set of d features $F = \{f_1, f_2, \dots, f_d\}$, set of selected features $S = \emptyset$.

- Step 2: *Compute MI*

Compute the MI of features $I(f_i; \Omega) \quad \forall i=1..d, \quad f_i \in F$ using equation (2.33), where the conditional entropy $H(\Omega|X_i)$ of feature $f_i=\{X_i\}$ is estimated using equation

$$\hat{H}(\Omega | X_i) = -\frac{1}{n} \sum_{j=1}^n \sum_{\omega=1}^{N_\omega} \hat{p}(\omega | x_{i,j}) \log_2 \hat{p}(\omega | x_{i,j}), \quad (2.37)$$

where the conditional entropy in equation (2.34) is estimated by replacing the integration with a summation of the sample points and assuming each sample has the same probability; the conditional probability density $p(\omega|x_i)$ is estimated using Parzen Window that employs univariate Gaussian kernel and normal optimal smoothing given in equation (2.36); and $\{x_{i,1}, x_{i,2}, \dots, x_{i,j}, \dots, x_{i,n}\} \in X_i$ comprises the data for feature f_i .

- Step 3: *Select the first feature*

Select the feature f_i that maximizes $I(f_i; \Omega)$ computed in step 2.

- Step 4: *Greedy selection*

Repeat

- a. Compute $I(f_i \cup S; \Omega) \quad \forall i=1..d, f_i \in F$, the joint MI between the feature i and selected features S with the output class where the conditional entropy is estimated using equation

$$\hat{H}(\Omega | \mathbf{X}) = -\frac{1}{n} \sum_{j=1}^n \sum_{\omega=1}^{N_\omega} \hat{p}(\omega | \mathbf{x}_j) \log_2 \hat{p}(\omega | \mathbf{x}_j), \quad (2.38)$$

where the conditional entropy in equation (2.34) is estimated by replacing the integration with a summation of the sample points and assuming each sample has the same probability; the conditional probability density $p(\omega|\mathbf{x})$ is estimated using Parzen Window that employs multivariate Gaussian kernel and normal optimal smoothing given in equation (2.36); and $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_j, \dots, \mathbf{x}_n\} \in \mathbf{X}$ comprises the data for features $f_i \cup S$.

b. Select next feature using equation

$$F = F - \{f_i\}, S = S \cup \{f_i\} |$$

$$I(S \cup \{f_i\}; \Omega) = \max_{j=1..d, f_j \in F} I(S \cup \{f_j\}; \Omega), \quad (2.39)$$

where F is the set of features to be selected from and S is the set of features selected.

Until $(|S| = k)$

The terminating condition in step 4 relies on the specification on the number of features to select in the parameter k . An autonomous method of stopping the greedy selection is to detect a negligible change in MI in the selection of the next feature f_i in equation

$$I(f_i \cup S) - I(S) \approx 0. \quad (2.40)$$

2.4.2.3 MI-based Best Individual Feature Parzen Window (MIBIFPW) algorithm

- Steps 1&2: *Initialize and compute MI*

Same as steps 1&2 of the MIPW algorithm.

- Step 3: *Select the best k features*

Repeat

Select the feature f_i that maximizes $I(f_i; \Omega)$ using equation

$$F = F - \{f_i\}, S = S \cup \{f_i\} | I(f_i; \Omega) = \max_{j=1..d, f_j \in F} I(f_j; \Omega), \quad (2.41)$$

where F is the set of features to be selected from and S is the set of features selected.

Until $|S| = k$

The MIBIFPW algorithm's overall structure is based on the *Mutual Information Feature Selection* (MIFS) algorithm in (Battiti, 1994), but differs in the computation of the MI in which the conditional entropy in equation (2.34) is estimated using Parzen Window. The MIPW or the MIBIFPW can be used to perform feature selection for any classification algorithms, since both are based on the filter approach.

2.4.3 Reasoning methods

A reasoning method, or equivalently, an approximate reasoning method, is a process by which a possibly imprecise conclusion is deduced from a collection of imprecise premises (Zadeh, 1979). Hence, the reasoning method in neuro-fuzzy systems defines the mathematical operations that are used to perform inference upon the collection of if-then fuzzy rules and given facts to derive a conclusion (Jang et al., 1997). There are three general approaches in the use of neural network techniques to perform fuzzy inference in linguistic neuro-fuzzy systems (Zhou & Quek, 1996):

- *Compositional Rule of Inference* method (CRI) that is originally proposed by Professor Lotfi Zadeh (1973),
- *Truth Value Restriction* method (TVR) (de Mántaras, 1990), and
- *Approximate Analogical Reasoning Schema* (AARS) (Turksen & Zhong, 1990).

TVR and AARS correspond closely to neural network implementations because the layers of a neural network effectively compute the similarity of input data with stored information and modify the output accordingly (Keller & Tahani, 1992). Although CRI is more complex, it plays the same role in approximate reasoning as *modus ponens* and *modus tollens* do in exact reasoning (Lee & Zhu, 1995). Thus many existing fuzzy reasoning methods are based on CRI (Yeung & Tsang, 1997). All three reasoning methods are used in the POPFNN family of networks. In 1996 and 1999, Zhou Rouwei and Chai Quek (1996; 1999) introduced POPFNN-TV and POPFNN-AARS and in 2003, POPFNN-CRI was introduced (Ang et al., 2003). The mathematical operations of these three reasoning methods are described in Section 2.4.4. For comparative study on the various reasoning methods, please refer to (Mizumoto & Zimmermann, 1982; Nakanishi, Turksen, & Sugeno, 1993; Yeung & Tsang, 1997).

2.4.4 Pseudo Outer-Product based Fuzzy Neural Network

This work employs the POPFNN using CRI and Singleton fuzzifier (POPFNN-CRI(S)) (Ang et al., 2003) as the foundation for the development of the proposed neuro-fuzzy system. Hence, this section reviews the POPFNN in more detail.

The POPFNN architecture is a five-layer neural network as shown in Figure 2-7. For simplicity, only the interconnections for the output y_m are shown. Each layer in POPFNN performs a specific fuzzy operation, so the nodes and operations of each layer are noted with a superscript of I to V for clarity. The inputs and outputs of the POPFNN are represented as non-fuzzy vector $\mathbf{X} = [x_1, x_2, \dots, x_i, \dots, x_{n_1}]$ and non-fuzzy vector $\mathbf{Y} = [y_1, y_2, \dots, y_m, \dots, y_{n_5}]$ respectively. The fuzzification of the inputs and the defuzzification of the outputs are respectively performed by the input and output linguistic layers respectively while the reasoning mechanism is collectively performed by the condition, rule and consequence layers.

Figure 2-7: Architecture of POPFNN

The numbers of neurons in the condition, rule and consequence layers are defined in equations (2.42)-(2.44) respectively.

$$n_2 = \sum_{i=1}^{n_1} J_i, \quad (2.42)$$

$$n_3 = \left[\prod_{i=1}^{n_1} J_i \right], \quad (2.43)$$

$$n_4 = \sum_{m=1}^{n_5} L_m, \quad (2.44)$$

where J_i is number of linguistic terms for the i^{th} input; L_m is the number of linguistic terms for the m^{th} output; n_1 is the number of inputs; n_2 is the number of neurons in the condition layer; n_3 is the number of if-then fuzzy rules or rule-based neurons; n_4 is the number of linguistic terms for the output; and n_5 is the number of outputs.

A detailed description of the functionality of each layer is given as follows:

- *Input linguistic layer* – contains neurons that are called input nodes. Each input node I_i^I represents an input linguistic variable (Zadeh, 1975) of the corresponding input x_i . Each node of the POPFNN with singleton fuzzifier transmits the crisp input directly to the condition layer. The POPFNN with non-singleton fuzzifier transmits the fuzzy input. For simplicity, only the POPFNN with singleton fuzzifier is described. The net input and output of an input node of POPFNN with singleton fuzzifier are defined in equation

$$\begin{aligned} \text{net input: } f_i^I &= x_i, \text{ and} \\ \text{net output: } o_i^I &= f_i^I, \end{aligned} \quad (2.45)$$

where x_i is the crisp value of the i^{th} input.

- *Condition layer* – contains neurons that are called input-label nodes. Each input-label node $IL_{i,j}^I$ represents the j^{th} linguistic term of the i^{th} input node from the input

layer. The input-label nodes constitute the antecedent of the if-then fuzzy rules. Each input-label node is represented by a membership function $\mu_{i,j}(x)$ where Gaussian or trapezoidal type of membership functions can be used. The function of an input-label node is the same for POPFNN-CRI(S) (Ang et al., 2003), POPFNN-TVR(S) (Zhou & Quek, 1996) and POPFNN-AARS(S) using disconsistency measure (Quek & Zhou, 1999). The net-input and net-output of an input-label node from these POPFNNs using a trapezoidal or Gaussian membership function are

$$\begin{aligned} \text{net input: } f_{i,j}'' &= o_i^I, \text{ and} \\ \text{net output: } o_{i,j}'' &= \begin{cases} \mu_T(o_i''; \alpha_{i,j}'', \beta_{i,j}'', \gamma_{i,j}'', \delta_{i,j}'') & \text{if trapezoidal} \\ \mu_G(o_i''; \alpha_{i,j}'', \beta_{i,j}'', \gamma_{i,j}'', \delta_{i,j}'') & \text{if Gaussian} \end{cases}, \end{aligned} \quad (2.46)$$

where μ_T is given in equation (2.4); μ_G is given in equation (2.5); $\alpha_{i,j}'', \delta_{i,j}'', \beta_{i,j}'', \gamma_{i,j}''$ are the parameters of the trapezoidal or Gaussian membership function for input-label node $IL_{i,j}''$; and o_i^I is the output of i^{th} input node I_i^I .

The function of an input-label node is different for POPFNN-AARS(S) using Hausdorff measure (Quek & Zhou, 1999) which uses only the centroid of the membership function. The net-input and net-output of an input-label node of POPFNN-AARS(S) using Hausdorff measure are

$$\begin{aligned} \text{net input: } f_{i,j}'' &= |o_i^I - v_{i,j}''|, \text{ and} \\ \text{net output: } o_{i,j}'' &= \frac{1}{(1 + f_{i,j}'')}, \end{aligned} \quad (2.47)$$

where $v_{i,j}'' = \frac{1}{2}(\beta_{i,j}'' + \gamma_{i,j}'')$ is the centroid of trapezoidal membership function;

$v_{i,j}'' = \frac{1}{2}(\beta_{i,j}'' + \delta_{i,j}'')$ is the centroid of the Gaussian membership function for input-

label node $IL_{i,j}''$; and o_i^I is the output of i^{th} input node I_i^I .

- *Rule-base layer* – contains neurons that are called rule nodes. Each rule node R_k^{III} represents an if-then fuzzy rule. The function of a rule node is the same for POPFNN-CRI(S) and POPFNN-TVR(S). The net input and output of a rule node from these POPFNNs are

$$\begin{array}{l} \text{POPFNN-TVR(S)} \quad \text{net input: } f_k^{III} = \min_i(o_{i,j}^{II}), \text{ and} \\ \text{POPFNN-CRI(S)} \quad \text{net output: } o_k^{III} = f_k^{III}, \end{array} \quad (2.48)$$

where $o_{i,j}^{II}$ is the output of the input-label node $IL_{i,j}^{II}$ that forms the antecedent conditions for the i^{th} input to the k^{th} rule node R_k^{III} .

The function of a rule node is different for POPFNN-AARS(S) when using different forms of the modification function (Quek & Zhou, 1999). The net input and output of a rule node from POPFNN-AARS(S) using expansion and reduction forms are

$$\begin{array}{l} \text{POPFNN-AARS(S) expansion form:} \quad \text{net input: } f_k^{III} = \min_i(o_{i,j}^{II}), \text{ and} \\ \text{net output: } o_k^{III} = \begin{cases} 1/f_k^{III} & f_k^{III} \geq \tau \\ 0 & \text{otherwise} \end{cases}, \end{array} \quad (2.49)$$

$$\begin{array}{l} \text{POPFNN-AARS(S) reduction form:} \quad \text{net input: } f_k^{III} = \min_i(o_{i,j}^{II}), \text{ and} \\ \text{net output: } o_k^{III} = \begin{cases} f_k^{III} & f_k^{III} \geq \tau \\ 0 & \text{otherwise} \end{cases}, \end{array} \quad (2.50)$$

where τ is a threshold value.

- *Consequence layer* – contains neurons that are called output-label nodes. The output-label node $OL_{m,l}^{IV}$ represents the l^{th} linguistic term of the output y_m . The output-label nodes constitute the conclusions of the if-then fuzzy rules. Each output-label node is represented by a membership function $\mu_{m,l}(x)$ similar to the input-label node where Gaussian or trapezoidal membership functions can be used. The function of an output-label node is different for each POPFNN using a

different inference scheme. The net input and output of the output-label node of POPFNN-CRI(S), POPFNN-TVR(S) and POPFNN-AARS(S) are given in equations (2.51)-(2.53). A choice of operators Σ or $\cup$ is selectable in equation (2.53) in which the mathematical operation of the latter is equivalent to a max operation.

$$\begin{aligned} \text{POPFNN-CRI(S):} \quad & \text{net input: } f_{m,l}^{IV} = \max_k(o_k^{III}), \text{ and} \\ & \text{net output: } o_{m,l}^{IV} = f_{m,l}^{IV}, \end{aligned} \quad (2.51)$$

$$\begin{aligned} \text{POPFNN-TVR(S):} \quad & \text{net input: } f_{m,l}^{IV} = \sum_k(o_k^{III}), \text{ and} \\ & \text{net output: } o_{m,l}^{IV} = \min(1, f_{m,l}^{IV}), \end{aligned} \quad (2.52)$$

$$\begin{aligned} \text{POPFNN-AARS(S):} \quad & \text{net input: } f_{m,l}^{IV} = \left(\sum_k(o_k^{III}), \max_k(o_k^{III}) \right), \text{ and} \\ & \text{net output: } o_{m,l}^{IV} = f_{m,l}^{IV}, \end{aligned} \quad (2.53)$$

where o_k^{III} is the output of the rule node R_k^{III} whose consequence is $OL_{m,l}^{IV}$.

- *Output linguistic layer* – contains neurons that are called output nodes. The output node O_m^V represents the output linguistic variable of the output y_m . The function of an output node is to perform defuzzification where each conclusion expressed in terms of fuzzy sets in the consequence layer is converted to a single real number. Three defuzzification methods are predominant in the fuzzy neural literature (Klir & Yuan, 1995): *centre of area*, *centre of maxima* and *mean of maxima*. The net input and output of the output node O_m^V of POPFNN-CRI(S), POPFNN-TVR(S) and POPFNN-AARS(S) using *centre of area* defuzzification are

$$\begin{aligned} \text{net input: } f_m^V &= \sum_{l=1}^{L_m} (o_{m,l}^{IV} \times v_{m,l}^{IV} \times w_{m,l}^{IV}), \text{ and} \\ \text{net output: } o_m^V &= \frac{f_m^V}{\sum_{l=1}^{L(m)} (o_{m,l}^{IV} \times w_{m,l}^{IV})}, \end{aligned} \quad (2.54)$$

where $v_{m,l}^{IV} = \frac{1}{2}(\beta_{m,l}^{IV} + \gamma_{m,l}^{IV})$ or $v_{m,l}^{IV} = \frac{1}{2}(\beta_{m,l}^{IV} + \delta_{m,l}^{IV})$ is the centroid of a trapezoidal or Gaussian membership function respectively for the output-label node $OL_{m,l}^{IV}$; and $w_{m,l}^{IV} = \frac{1}{2}((\delta_{m,l}^{IV} - \gamma_{m,l}^{IV}) - (\beta_{m,l}^{IV} - \alpha_{m,l}^{IV}))$ or $w_{m,l}^{IV} = (\alpha_{m,l}^{IV} + \gamma_{m,l}^{IV} + \delta_{m,l}^{IV} - \beta_{m,l}^{IV})$ is the width of the trapezoidal or Gaussian membership function respectively.

2.4.4.1 Learning process

POPFNN uses the hybrid learning approach. In the first stage, the clustering approach is used to generate the membership functions by employing the Learning Vector Quantization (LVQ) algorithm (Kohonen, 1989) or the Fuzzy Kohonen Partitioning (FKP) algorithm (Ang et al., 2003). The former is described in Section 2.4.1.2 and the latter is an improvement based on the former. After the membership functions of the condition and consequent layers are generated, the Pseudo Outer-Product (POP) algorithm described in Section 2.4.4.2 or its variant, the LazyPOP (Quek & Zhou, 2001), is employed to identify the if-then fuzzy rules. In the final stage, supervised learning using backpropagation is employed to optimize the membership functions in the condition and consequent layers.

Figure 2-8: POP in the identification of Output Label nodes for k^{th} rule node

Figure 2-8 shows the learning process of the POP algorithm in the identification of the output-label nodes for rule node R_k^{III} . Each rule node R_k^{III} is linked to only one input-label node from each input node and only one output-label node from each output node. The links of rule nodes to the antecedent in the condition layer and the consequent in the consequence layer are mathematically denoted as sets $(\mathbf{C}^I, \mathbf{D}^V)$. The antecedent of rule nodes is represented by a set $\mathbf{C}^I = \{c_1, c_2, \dots, c_i, \dots, c_{n_i}\}$ where c_i is a condition variable such that $c_i \in \mathbf{IL}_i^{II}$. The set of condition labels c_i can take is semantically represented by the set of input-label nodes $\mathbf{IL}_i^{II} = \{IL_{i,1}^{II}, IL_{i,2}^{II}, \dots, IL_{i,j}^{II}, \dots, IL_{i,J_i}^{II}\}$ but a computational notation of $\mathbf{IL}_i^{II} = \{1, 2, \dots, j, \dots, J_i\}$ is used. Similarly, the consequent of rule nodes is represented by a set $\mathbf{D}^V = \{d_1, d_2, \dots, d_m, \dots, d_{n_s}\}$ where d_m is a consequent variable such that $d_m \in \mathbf{OL}_m^{IV}$. The set of consequent labels d_m can take is semantically represented by the set of output-label nodes $\mathbf{OL}_m^{IV} = \{OL_{m,1}^{IV}, OL_{m,2}^{IV}, \dots, OL_{m,l}^{IV}, \dots, OL_{m,L_m}^{IV}\}$ but a computational notation of $\mathbf{OL}_m^{IV} = \{1, 2, \dots, l, \dots, L_m\}$ is used. The specific links of a rule node R_k^{III} to the antecedent in the condition layer and the consequent in the consequence layer is denoted as $(\mathbf{C}_k^I, \mathbf{D}_k^V)$. During POP learning, the consequent of rule node R_k^{III} are computationally represented as weights w_k^{III} .

2.4.4.2 Pseudo Outer-Product (POP) algorithm

- Step 1: *Rule initialization*

Initialize the links $\mathbf{C}_k^I$ of rule nodes to input-label nodes for $k=1..n_3$ such that all possible combination of if-then fuzzy rules are created using (2.58). Initialize $w_{k,m,l}^{III}=0$ for $k=1..n_3, m=1..n_5, l=1..L_m$.

- Step 2: *Hebbian learning*

Given the training data $\{\{\mathbf{X}_1, \mathbf{Y}_1\}, \{\mathbf{X}_2, \mathbf{Y}_2\}, \dots, \{\mathbf{X}_p, \mathbf{Y}_p\}, \dots, \{\mathbf{X}_n, \mathbf{Y}_n\}\}$,

determine weights $w_{k,m,l}^{III}$ using Pseudo Outer Product Hebbian learning (Quek & Zhou, 2001; Hebb, 1949) as given in equation (2.55).

For $p=1..n$, $m=1..n_5$:

$$w_{k,m,l}^{III} = \sum_{p=1}^{n_3} f_{k,p}^{III} \mu_{m,l} (y_{m,p}) \quad (2.55)$$

where $w_{k,m,l}^{III}$ is the weight of the link between R_k^{III} and $OL_{m,l}^{IV}$; $f_{k,p}^{III}$ is the firing strength of the p^{th} instant of training data; $\mu_{m,l}$ is the membership value of the l^{th} output-label node for the m^{th} output; and $y_{m,p}$ is the m^{th} element of the p^{th} instant of $\mathbf{Y}$.

End For

- Step 3: *Rule formulation*

Determine the link of each rule node R_k^{III} to output-label nodes, denoted as $\mathbf{D}_k^V = \{d_1, d_2, \dots, d_m, \dots, d_{n_5}\}$, where $d \in \mathbf{OL}_m^{IV}$ and $\mathbf{OL}_m^{IV} = \{1, 2, \dots, l, \dots, L_m\}$.

For $k=1..n_3$, $m=1..n_5$:

a. Compute the consequence link with the highest weight using equation

$$w_{k,m,l_{\max}}^{III} = \max_l w_{k,m,l}^{III} \quad (2.56)$$

b. Select or delete the link to the consequence of m^{th} output using equation

$$d_m = \begin{cases} l_{\max} & \text{if } w_{k,m,l_{\max}}^{III} > 0 \\ \emptyset & \text{if } w_{k,m,l_{\max}}^{III} = 0 \end{cases} \quad (2.57)$$

c. Delete rule node R_k^{III} if $\mathbf{D}_k^V = \emptyset$, that is $\forall d_m \in \mathbf{D}_k^V, d_m = \emptyset$.

End for

d. Update n_3 with the total number of remaining if-then fuzzy rules.

The initialization of if-then fuzzy rules in step 1 and the computation of $f_{k,r}^{III}$ in step 2 involve an integer rule index k . The computation of k given $\mathbf{C}_k^I$ is given in equation (2.58). This equation is constructed such that each condition variable $c_i \in \mathbf{C}_k^I$ construes a digit in a positional number system (Knuth, 1997) where c_1 is the least significant digit and c_{n_i} is the most significant digit. As the set of each condition labels c_i can take is $\mathbf{IL}_i^{II} = \{1, 2, \dots, J_i\}$, the term $\prod_{p=0}^{i-1} J_p$ computes the associated weight of each digit and the addition and subtraction of unity in the equation ensures that all possible combinations of $\mathbf{C}_k^I$ falls within the contiguous integer range of $k=1..n_3$.

$$k = \sum_{i=1}^{n_i} \left((c_i - 1) \times \prod_{p=0}^{i-1} J_p \right) + 1, \quad J_0 = 1 \quad (2.58)$$

where J_i is the number of linguistic terms for the i^{th} input; c_i is the condition variable of the i^{th} input of if-then fuzzy rule R_k such that $c_i \in \mathbf{C}_k^I$, $c_i \in \mathbf{IL}_i^{II}$; $\mathbf{C}_k^I$ is the set of condition variables of if-then fuzzy rule R_k , $\mathbf{C}_k^I = \{c_1, c_2, \dots, c_i, \dots, c_{n_i}\}$; $\mathbf{IL}_i^{II}$ is the set of condition labels c_i can take, $\mathbf{IL}_i^{II} = \{1, 2, \dots, J_i\}$.

2.4.5 Historical notes

Table 2-3 lists some of the research development in neuro-fuzzy systems in chronological order. For more comprehensive reviews, please refer to (Buckley & Hayashi, 1994; Buckley & Yoichi Hayashi, 1995; Lin & Lee, 1996; Mitra & Hayashi, 2000; Takagi, 2000; Babuška & Verbruggen, 2003; Nédjah & Mourelle, 2005).

Table 2-3: List on some development of neuro-fuzzy systems in chronological order

1991	Hideyuki Takagi and Isao Hayashi (1991) pioneered the work in neuro-fuzzy systems with <i>nn-driven fuzzy reasoning</i> . Chin-Teng Lin and George Lee (1991) introduced <i>Fuzzy Adaptive Learning Control Network</i> (FALCON).
1992	James Keller, Ronald Yager and Hossein Tahani (1992; Keller & Tahani, 1992) proposed fuzzy logic inference by feed-forward neural networks. Saman Halgamuge and Manfred Glesner (1992; 1994) developed the <i>neuro-fuzzy model</i> (FeNe-I). Isao Hayashi et al. (Hayashi et al., 1992) developed the <i>Neural Network Driven Fuzzy Reasoning</i> (NNDFR) model. Hamid Berenji and Pratap Khedkar (1992) developed the <i>Approximate Reasoning-based Intelligent Control</i> (ARIC) and <i>generalized ARIC</i> (GARIC). Gail Carpenter et al. (1992) developed the <i>fuzzy ARTMAP</i> . Bart Kosko (1992) wrote the first book on neural networks and fuzzy systems.
1993	Roger Jang (1993) developed the <i>Adaptive-Network-based Fuzzy Inference Systems</i> (ANFIS), one of the first hybrid neuro-fuzzy systems for function approximation based on the TSK model.
1996	Chin-Teng Lin and George Lee (1996) wrote a comprehensive book on neuro-fuzzy synergism. Rouwei Zhou and Chai Quek (1996) developed the <i>Pseudo Outer-Product based Fuzzy Neural Networks</i> using <i>Truth Value Restriction</i> (POPFNN-TVR) based on the Mamdani model.
1997	Cheng-Jian Lin and Chin-Teng Lin (1997) introduced a five-layer feed-forward fuzzy neural network that employed fuzzy ART clustering technique. Detlef Nauck, Frank Klawonn and Rudolf Kruse (1997) developed <i>NEFCON</i> , <i>NEFCLASS</i> and <i>NEFPROX</i> for neuro-fuzzy control, classification and approximation respectively.
1998	Nikola Kasabov (1998; 2001) developed the <i>Evolving Fuzzy Neural Network</i> (EFuNN). Chia-Feng Juang and Chin-Teng Lin (1998) developed the <i>Self Constructing Neural Fuzzy Inference Network</i> (SONFIN) based on the TSK model.
1999	Chai Quek and Rouwei Zhou (1999) developed the <i>POPFNN</i> using the <i>Approximate Analogical Reasoning Schema</i> (POPFNN-AARS).
2002	Nikola Kasabov and Qun Song (2002) developed the <i>Dynamic Evolving Neural-Fuzzy Inference System</i> (DENFIS) based on the TSK model. Whye Loon Tung and Chai Quek (2002b) proposed the <i>Generic Self-Organizing Fuzzy Neural Network</i> (GenSoFNN) based on the Mamdani model.
2003	Leszek Rutkowski and Krzysztof Cpałka (2003) introduced <i>flexible neuro-fuzzy systems</i> . <i>POPFNN</i> using the <i>Compositional Rule of Inference</i> (POPFNN-CRI) was developed (Ang et al., 2003).
2005	Qun Song and Nikola Kasabov (2005) developed <i>Neural Fuzzy Inference</i> (NFI) for transductive reasoning.
2007	Chunshien Li and Kuo-Hsiang Cheng (2007) proposed a <i>hybrid-learning based recurrent neuro-fuzzy system</i> using six layers of neural network.

2.4.6 Advantages and issues in modeling data

The advantages of modeling data using neuro-fuzzy systems are:

1. They are able to abstract humanly interpretable if-then fuzzy rules from available numerical data (Mitra & Hayashi, 2000; Babuška & Verbruggen, 2003).
2. They are universal approximators (Feuring & Lippe, 1999; Wang & Wei, 2000; Ying et al., 1999; Tikk et al., 2003) similar to neural networks and fuzzy systems.
3. They are non-linear models capable of non-linear system identification (Babuška & Verbruggen, 2003) similar to neural networks and fuzzy systems.
4. They are gray-box models (Sjoberg et al., 1995; Lindskog, 1997) in which the modeling can be performed using a priori knowledge from human experts as well as the available numerical data.

The prevailing issues of modeling data using neuro-fuzzy systems are:

1. They are unable to abstract models of high interpretability and accuracy (Casillas et al., 2003). To yield high accuracy, they employ unconstrained optimization of membership functions, which are derailed from human-interpretable terms (de Oliveira, 1999b); or employ a large number of if-then fuzzy rules (Ying et al., 1999), which exceeded human interpretability (Tikk & Baranyi, 2003). To yield high interpretability, heuristic thresholds are used to identify fewer if-then fuzzy rules or perform pruning, but accuracy is sacrificed (Moser, 1999).
2. They employ feature selection based on filter approach to model high dimensional data, but are inferior to the wrapper approach (Kohavi & John, 1997).
3. Their users have to deal with a number of potentially necessary parameters in the neural learning algorithms that generate membership functions and identify the if-then fuzzy rules. Thus, their use requires specialized skills and knowledge since it is not a fully automated process (Babuška & Verbruggen, 2003).

2.5 Rough Set Theory

Professor Zdzislaw Pawlak (1982; 1991; 2002) introduced rough set theory as an extension of classical set theory for the design of intelligent systems characterized by incomplete or vague information. It is different from and complementary to other generalizations of classical set theory such as fuzzy sets and multisets (Yao, 1996), since it is based on the concept that every object is associated with some information and that the objects that are characterized by the same information are indiscernible.

2.5.1 Upper and lower approximation

A rough set approximates a vague concept using a pair of precise concepts, called the lower and upper approximations. Given a universal set $\mathbf{U}$, let R be an equivalence relation on $\mathbf{U}$. A rough set $R(A)$ of a given set $A \in \mathbf{U}$ is represented by a tuple that comprises of two subsets given in equation (2.59).

$$R(A) = \langle \underline{R}(A), \bar{R}(A) \rangle, \quad (2.59)$$

where $\underline{R}(A)$ is the *lower approximation* of A given in equation (2.60) and $\bar{R}(A)$ is the *upper approximation* of A given in equation (2.61).

$$\underline{R}(A) = \{x \in \mathbf{U} \mid [x]_R \subseteq A\}, \quad (2.60)$$

$$\bar{R}(A) = \{x \in \mathbf{U} \mid [x]_R \cap A \neq \emptyset\}, \quad (2.61)$$

where $[x]_R$ denotes the equivalence class in $\mathbf{U}/R$ that contains $x \in \mathbf{U}$, and $\mathbf{U}/R$ denotes the family of all equivalence class induced on $\mathbf{U}$ by R .

The *lower approximation* describes domain objects that certainly belong to the subset of interest, whereas the *upper approximation* describes domain objects that possibly belong to the subset of interest.

Table 2-4: An example of an information system adopted from (Quilan, 1983)

Object	Height	Hair	Eyes	Classification
o_1	short	blond	blue	+
o_2	short	blond	brown	-
o_3	tall	red	blue	+
o_4	tall	dark	blue	-
o_5	tall	dark	blue	-
o_6	tall	blond	blue	+
o_7	tall	dark	brown	-
o_8	short	blond	brown	-

Consider the information system in Table 2-4, each object is described by three attributes: height, hair and eyes. Using attribute $A=\{\text{Hair}\}$, the universe is partitioned into the equivalence classes $\{o_1, o_2, o_6, o_8\}$, $\{o_3\}$ and $\{o_4, o_5, o_7\}$ that correspond to blond, red and dark respectively. With respect to the class + which consists of $\{o_1, o_3, o_6\}$, the lower approximations is $\underline{R}_A(+)=\{o_3\}$ and the upper approximation is $\overline{R}_A(+)=\{o_1, o_2, o_3, o_6, o_8\}$ respectively.

2.5.2 Knowledge reduction

Rough Set theory also provides rigorous mathematical techniques for discovering regularities in data (Pawlak, 1991). Two fundamental concepts of rough set knowledge reduction are relevant in the identification of if-then fuzzy rules; namely, a *reduct* and the *core*. A *reduct* of knowledge is its essential part, which suffices to define all basic concepts occurring in the considered knowledge, whereas the *core* is the most important part of the knowledge. Central to this rough set approach is the concept of attribute dispensability in terms of indiscernibility relations (Pawlak, 1991). An attribute $R \in \mathbf{R}$ is dispensable if it satisfies

$$\text{IND}(\mathbf{R}) = \text{IND}(\mathbf{R} - \{R\}), \quad (2.62)$$

where $\text{IND}(\mathbf{P})$ is the indiscernibility relation over $\mathbf{P}$, which is the intersection of all equivalence relations belonging to $\mathbf{P}$ given in equation (2.63).

$$\text{INP}(\mathbf{P}) = \bigcap_{R \in \mathbf{P}} [x]_R \quad (2.63)$$

The rough set knowledge reduction uses the value-attribute pair to represent the knowledge of decision rules (Pawlak, 1991). When decision rules are represented as such, rough set theory then supplies the logical methods employing attribute dispensability and decision rule consistency for knowledge reduction and analysis. As decision rules do not have the attributes of truth or falsity as opposed to formulas, consistency and inconsistency are used instead to describe decision rules in computational terms. The following sections elaborate more on the knowledge representation system (KRS), attribute dispensability and consistency from rough set theory.

2.5.2.1 Knowledge representation system

To express mathematically how rough set theory is applied in knowledge reduction, the Tarski's style semantics of the Decision Logic language (DL-language) is used in (Pawlak, 1991) where the notions of a *model* and *satisfiability* are employed. Knowledge is represented as a model $S=(\mathbf{U},\mathbf{A})$ where $\mathbf{U}$ is the universe of discourse and $\mathbf{A}$ is the set of attributes. The formula S is satisfied by an element x in $\mathbf{U}$ if and only if an extraction of attribute value from x belongs to the set of attribute $\mathbf{A}$. In other words, an object $x \in \mathbf{U}$ satisfies S if and only if $a(x)=v$ where $a \in \mathbf{A}$ and $v \in \mathbf{V}$, $\mathbf{V} = \bigcup \mathbf{V}_a$ and $\mathbf{V}_a$ is the set of attribute values for $a \in \mathbf{A}$. An atomic formula is represented as (a,v) in DL-language for any $a \in \mathbf{A}$ and $v \in \mathbf{V}$. A formula is expressed in the form

$$(a_1, v_1) \wedge (a_2, v_2) \wedge \cdots \wedge (a_i, v_i) \cdots \wedge (a_n, v_n), \quad (2.64)$$

where v_i is the attribute constant value, $v_i \in \mathbf{V}_{a_i}$; a_i is the attribute constant, $a_i \in \mathbf{A}$; and $\wedge$ is the conjunction connective.

In a decision rule system, $\mathbf{A}=\{\mathbf{C}\cup\mathbf{D}\}$ where $\mathbf{C}$ is the set of conditional attributes and $\mathbf{D}$ is the set of decision attributes. If $\mathbf{P}=\{a_1, a_2, \dots, a_i, \dots, a_n\}$ and $\mathbf{P}\subseteq\mathbf{A}$, then the formula in equation (2.64) is called a P-basic formula or in short P-formula. The disjunction of all P-formulas satisfied in the system S is represented as $\sum_S \mathbf{P}$. If $\mathbf{P}=\mathbf{A}$, then $\sum_S \mathbf{A}$ is called the characteristic formula of S . The following example illustrates how the knowledge is represented.

Table 2-5: An example of the Knowledge Representation System (KRS) using attribute-value table adopted from (Pawlak, 1991)

U	a_1	a_2	a_3
1	1	0	2
2	2	0	3
3	1	1	1
4	1	1	1
5	2	1	3
6	1	0	3

Expressing the atomic formula (a_i, v) in short as $a_{i,v}$ and omitting the conjunction symbol $\wedge$, the characteristic formula of the knowledge in Table 2-5 is

$$\sum_S \mathbf{A} = a_{1,1}a_{2,0}a_{3,2} \vee a_{1,2}a_{2,0}a_{3,3} \vee a_{1,1}a_{2,1}a_{3,1} \vee a_{1,2}a_{2,1}a_{3,3} \vee a_{1,1}a_{2,0}a_{3,3}. \quad (2.65)$$

Now given a decision rule $\phi \rightarrow \psi$ where ϕ and ψ are C-basic and D-basic formula respectively, then the decision rule $\phi \rightarrow \psi$ is called a CD-basic decision rule. Considering the knowledge in Table 2-5, assume that $\mathbf{C}=\{a_1, a_2\}$ and $\mathbf{D}=\{a_3\}$, sets $\mathbf{C}$ and $\mathbf{D}$ associate the CD-basic decision rules given in equation (2.66). These CD-basic decision rules form the decision rule base denoted as $(\mathbf{C}, \mathbf{D})$.

$$\begin{aligned} a_{1,1}a_{2,0} &\rightarrow a_{3,2} \\ a_{1,2}a_{2,0} &\rightarrow a_{3,3} \\ a_{1,1}a_{2,1} &\rightarrow a_{3,1} \\ a_{1,2}a_{2,1} &\rightarrow a_{3,3} \\ a_{1,1}a_{2,0} &\rightarrow a_{3,3} \end{aligned} \quad (2.66)$$

2.5.2.2 Rule consistency

When a decision rule $\phi \rightarrow \psi$ satisfies S , the decision rule is consistent in S if and only if for any decision rule $\phi' \rightarrow \psi'$ in S , $\phi = \phi'$ implies $\psi = \psi'$ (Pawlak, 1991). Considering the knowledge in Table 2-5 with $\mathbf{C}=\{a_1, a_2\}$ and $\mathbf{D}=\{a_3\}$, rule $a_{1,1}a_{2,0} \rightarrow a_{3,2}$ is inconsistent with rule $a_{1,1}a_{2,0} \rightarrow a_{3,3}$. Thus, the decision rule base $(\mathbf{C}, \mathbf{D})$ is inconsistent. If either rule is removed, then the decision rule base $(\mathbf{C}, \mathbf{D})$ is consistent.

2.5.2.3 Attribute reduction

Considering a consistent decision rule base $(\mathbf{C}, \mathbf{D})$, an attribute $a_i \in \mathbf{C}$ is dispensable if and only if the decision rule base $((\mathbf{C} - \{a_i\}), \mathbf{D})$ is consistent (Pawlak, 1991). If all attributes $a_i \in \mathbf{C}$ are indispensable in $(\mathbf{C}, \mathbf{D})$, then the decision rule base $(\mathbf{C}, \mathbf{D})$ is independent. The concept of a *reduct* of knowledge in rough set theory is the subset of attributes $\mathbf{R} \subseteq \mathbf{C}$ where the decision rule base $(\mathbf{R}, \mathbf{D})$ is independent and consistent. The concept of the *core* of knowledge in rough set theory is the set of all indispensable attributes of the decision rule base $(\mathbf{C}, \mathbf{D})$, denoted as $CORE(\mathbf{C}, \mathbf{D})$.

2.5.2.4 Decision rule reduction

The concept of a reduct can also be applied to simplify dispensable conditions of decision rules similar to attribute reduction performed on the decision rule base (Pawlak, 1991). Considering a consistent CD-basic decision rule $\phi \rightarrow \psi$ and $\mathbf{Q} \subseteq \mathbf{C}$, let the notation $\phi / \mathbf{Q}$ denotes the Q-basic decision rule obtained by removing $(\mathbf{C} - \mathbf{Q})$ from ϕ . An attribute a_i is dispensable in the consistent CD-basic decision rule $\phi \rightarrow \psi$ if and only if the CD-basic decision rule $\phi / (\mathbf{C} - \{a_i\}) \rightarrow \psi$ is also consistent.

2.5.3 Historical notes

Table 2-6 lists some of the research development in rough set theory in chronological order. For more comprehensive reviews, please refer to (Pawlak, 2002; Jensen & Shen, 2004; Curry, 2003; Pawlak & Skowron, 2007c).

2.5.4 Advantages and issue in analyzing data

The advantages of using rough set theory in analyzing data are

1. It provides efficient and rigorous mathematical techniques for discovering patterns in data (Pawlak, 1991).
2. It does not require a priori information about data such as probability distributions in statistics, probability assignments in Dempster-Shafer theory, or membership functions in fuzzy set theory (Pawlak & Skowron, 2007c).

The prevailing issues of modeling data using rough set theory are:

1. It can only handle discretized attribute values and thus cannot handle crisp and real-valued attribute values in most data (Jensen & Shen, 2004).
2. It cannot handle attribute values that are similar but differ slightly as a result of noise, since it consider these attribute values as different values of a different order of magnitude (Jensen & Shen, 2004).
3. It cannot handle high dimensional data since there is a factorial increase in the number of potential reducts involving possible combinations of the conditional attributes (Skowron & Rauszer Cecylia, 1992; Curry, 2003).

Table 2-6: List on some development of rough set theory in chronological order

1982	Zdzislaw Pawlak (1982; 1991) pioneered the work in rough sets. He introduced the basic notion of rough sets and approximation space to deal with imprecision or vagueness.
1984	Zdzislaw Pawlak (1984) introduced the idea of approximate classification based on the concept of a rough set.
1990	Didier Dubois and Henri Prade (1990) introduced the idea of combining rough sets with fuzzy sets. (1992)
1992	Andrzej Skowron and Ceylia Rauszer (1992) developed <i>discernibility matrix</i> and <i>discernibility function</i> for efficient methods of computing the reducts, core and dependencies in solving knowledge reduction problems.
1993	Wojciech Ziarko (1993) developed variable precision rough sets to model classification problems involving uncertain or imprecise information.
1998	Yi Yu Yao (1998) reviewed and compared theories of fuzzy sets and rough sets. Pawan Lingras (1998) compared rough neurons and neofuzzy neurons. Marzena Kryszkiewicz (1998) presented the rough set approach in reasoning in incomplete information systems.
1997	Tsau Young Lin and Nick Cercone (1997) compiled a book on articles that used rough set theory for mining databases.
1999	Sankar Pal and Andrzej Skowron (1999) compiled a book on articles that combine fuzzy set theory with rough set theory in building intelligent systems in the soft computing paradigm.
2002	Qiang Shen and Alexios Chouchoulas (2002) developed a <i>fuzzy-rough approach</i> for generating classification rules. Robert Nowicki and Leszek Rutkowski (2002) developed <i>rough-neuro-fuzzy system</i> , which synergized the rough set method of handling incomplete information with neuro-fuzzy systems.
2003	Roman Swiniarski and Andrzej Skowron (2003) synergized rough set methods with principal component.
2005	<i>Rough set-based Pseudo Outer-Product</i> (RSPOP) algorithm, which applied the concept of knowledge reduction of rough set theory in abstracting if-then fuzzy rules in neuro-fuzzy systems, is proposed (Ang & Quek, 2005). This approach of synergizing rough-set theory with neuro-fuzzy system is named <i>rough set-based neuro-fuzzy approach</i> to distinguish from the rough-neuro-fuzzy approach in (Nowicki & Rutkowski, 2002).
2006	The basic concept of <i>rough set-based neuro-fuzzy system</i> is initially proposed (Ang & Quek, 2006a).
2007	Richard Jensen and Qiang Shen (2007a) developed fuzzy rough feature selection. Zdzislaw Pawlak and Andrzej Skowron (2007b; 2007a) presented some extension of the rough set approach and discussed methods of combining rough sets and Boolean reasoning with applications in pattern recognition, machine learning, data mining and conflict analysis.

2.6 Confidence in modeling classification data

The performance of a system in modeling data is mainly characterized by the accuracy of using the system in the predictions on unseen samples, in which errors are unavoidable due to uncertainties and noise. In modeling classification data, the inputs to the system are usually continuous variables and the output is the class that has discrete values. To increase the accuracy of systems in modeling classification data, classifiers employ a reject option to withhold classification decision if the confidence of the classification decision is not sufficiently high (Ha, 1997; De Stefano, Sansone, & Vento, 2000). Recently, there are a number of research efforts to control classifier performance using the reject option (Li & Sethi, 2006b; Delany, Cunningham, Doyle, & Zamolotskikh, 2005; Li & Sethi, 2006a; Yoma, Carrasco, & Molina, 2005; Kim & Ko, 2005). In 2006, Li and Sethi (2006b) coined the term confidence-based classifier, which classifies samples or rejects them according to the output scores of the classifier. Currently, existing neuro-fuzzy systems do not employ the reject option to increase the accuracy in modeling classification data.

2.6.1 Probabilistic rejection rule

Chow (1970) pioneered the design of classifiers with optimal rejection rates using the optimal probabilistic rejection rule that minimizes the reject rate for a given error rate

$$\text{reject } \mathbf{x} \text{ if } \frac{\max_{\omega_m \in \Omega} p(\omega_m | \mathbf{x})}{\sum_{\omega \in \Omega} p(\omega | \mathbf{x})} < 1 - t, \quad (2.67)$$

where $p(\omega | \mathbf{x})$ is the conditional probability of class ω given the data sample $\mathbf{x}$; class $\omega \in \Omega$ in which $\Omega = \{1, \dots, N_\omega\}$ contains the set of all possible classes out of a total of N_ω classes; class $\omega_m \in \Omega$ is one of the classes that yield the maximum conditional probability; and t is the rejection threshold. The optimal probabilistic classification rule is defined as equation

$$\text{decide } \omega_m \text{ if } \max_{\omega_m \in \Omega} p(\omega_m | \mathbf{x}) \geq 1 - t. \quad (2.68)$$

If the total number of classes $N_\omega=2$, the rejection threshold t takes a value between 0 and 0.5. When $t=0$, all samples will be rejected; when $t=0.5$, the resulting classification rule is one without the reject option; otherwise, any sample with the maximum posterior probability of less than $1-t$ will be rejected (Li & Sethi, 2006b). The significance of the optimal probabilistic rejection rule is that it empowers probabilistic classifiers with the option of rejecting samples below the rejection threshold so that the error rate is reduced compared to classifiers without the reject option (Li & Sethi, 2006b). The rejection threshold t can be determined if and only if the inherent probability distributions of the data and the desired error rate are known. However, when t is decreased, the error rate decreases but the rejection rate increases.

2.6.2 Possibilistic rejection rule

A linguistic neuro-fuzzy system, when employed in modeling classification data, is a possibilistic classifier that treats a numerical data sample $\mathbf{x}$ as a fuzzy linguistic variable χ that is associated with a possibility distribution $\pi_\chi(\mathbf{x})$ similar to the way a random variable $\mathbf{x}$ is associated with a probability distribution $p(\mathbf{x})$ (Denoex & Zouhal, 2001; Zadeh, 1978b). Hence, a possibilistic classifier estimates $\pi_{\omega|\chi}(\mathbf{x}, \omega)$ from the training data samples and predicts the class ω with the highest possibility similar to the way a probabilistic classifier estimates $p(\omega|\mathbf{x})$ from the training data samples and predicts the class ω with the highest probability. The difference between the possibilistic and the probabilistic approach is that the former handles the meaning of information using linguistic if-then rules in equation (2.69) while the later handles the measure of information (Zadeh, 1978b).

$$R_k: \text{if } \chi_1 \text{ is } X_{1,j} \text{ and } \chi_2 \text{ is } X_{2,j} \cdots \chi_d \text{ is } X_{d,j} \text{ then } \varpi \text{ is } Y_\omega, \quad (2.69)$$

where χ_i and ϖ are fuzzy linguistic variables that takes values in the set X_i and Y respectively; $X_{i,j}$ is a fuzzy subset representing the j^{th} linguistic label from a universal set X_i ; and Y_ω is the fuzzy singleton representing the class ω from a universal set Y . The if-then fuzzy rule R_k expresses a fuzzy relation $R_k(\mathbf{X}, Y)$ on $\mathbf{X} \times Y$ that restricts the possible values of Y given $\mathbf{X}$ which can be represented by the conditional possibility distribution function (de Mántaras, 1990)

$$\pi_{\varpi|\chi}(\mathbf{x}, \omega) = I_T [\mu_{\chi}(\mathbf{x}), \mu_Y(\omega)], \quad (2.70)$$

where $\pi_{\varpi|\chi}(\mathbf{x}, \omega)$ is the conditional possibility of $\varpi=\omega$ given $\chi=\mathbf{x}$ (de Mántaras, 1990); I_T is the fuzzy implication function associated with a triangular norm T ; μ_{χ} is the composed membership function of the proposition “ χ is $\mathbf{X}$ ” that characterized the composed possibility distribution of χ ; $\mu_Y(\omega)=1$ for class ω and $\mu_Y(\omega)=0$ otherwise.

The optimal probabilistic rejection and classification rules (Chow, 1970) in equations (2.67) and (2.68) can be extended to the possibilistic approach by replacing the conditional probability $p(\omega|\mathbf{x})$ of class ω given the data sample $\mathbf{x}$ in equations (2.67) and (2.68) with the conditional possibility $\pi_{\varpi|\chi}(\mathbf{x}, \omega)$ of $\varpi=\omega$ given $\chi=\mathbf{x}$. Thus the optimal possibilistic rejection rule which minimizes the reject rate for a given error rate is

$$\text{reject } \mathbf{x} \text{ if } \frac{\max_{\omega_m \in \Omega} \pi_{\varpi|\chi}(\mathbf{x}, \omega_m)}{\sum_{\omega \in \Omega} \pi_{\varpi|\chi}(\mathbf{x}, \omega)} < 1 - t, \quad (2.71)$$

where $\pi_{\varpi|\chi}(\mathbf{x}, \omega)$ is the conditional possibility of $\varpi=\omega$ given $\chi=\mathbf{x}$; class $\omega \in \Omega$ in which $\Omega = \{1, \dots, N_\omega\}$ contains the set of all possible classes out of a total of N_ω classes; class $\omega_m \in \Omega$ is one of the classes that yield the maximum conditional possibility; and t is the rejection threshold. The optimal possibilistic classification rule is thus

$$\text{decide } \omega \text{ if } \max_{\omega_m \in \Omega} \pi_{\omega|X}(\mathbf{x}, \omega_m) \geq 1 - t. \quad (2.72)$$

Similar to the optimal probabilistic rejection rule in equation (2.67), the rejection threshold t can be determined if and only if the inherent possibility distributions of the data and the desired error rate are known.

2.6.3 Advantages and issues in modeling classification data

The advantages of using the reject option in modeling classification data are:

1. It improves the accuracy of the system in classifying unseen samples by rejecting the classification of samples that fall near the decision boundary of two classes (Jain et al., 2000).
2. It also improves the reliability of the classification since samples near the decision boundary are classifications made with low confidence (Li & Sethi, 2006b).
3. The probabilistic rejection rule can be easily implemented using the output scores of classifiers (Li & Sethi, 2006b).

The main issues of using the reject option in modeling classification data are:

1. The output scores of most classifiers are not predictive of classification confidence (Delany et al., 2005). Classification confidence in this case refers to a measure that reflects the classification reliability (De Stefano et al., 2000; Cordella, De Stefano, Tortorella, & Vento, 1995). This measure is known as the *Confidence Measure* in automatic speech recognition literature (Yoma et al., 2005; Kim & Ko, 2005).
2. The accuracy of the system is increased by increasing the rejection threshold, but this also increases the number of rejected samples which includes samples that would have been correctly classified (De Stefano et al., 2000). This issue is known as the error-reject tradeoff in the literature (Jain et al., 2000; Chow, 1970; De Stefano et al., 2000).

2.7 Summary

This chapter reviews the modeling of data using neural networks, fuzzy systems, neuro-fuzzy systems, and rough set theory. In addition, the use of a reject option in modeling classification data is also reviewed. The advantages and issues of modeling data using each of these techniques are discussed. Neuro-fuzzy systems that synergize neural networks with fuzzy systems are reviewed in more detail. Specifically, the generation of membership functions, the identification of if-then fuzzy rules, and the reasoning methods are discussed. The Pseudo-Outer Product Fuzzy Neural Network, which is significant to this work, is also examined in more detail.

The prevailing issue in neuro-fuzzy systems is that they are unable to abstract models of high interpretability and accuracy. This issue is even more prominent in modeling high dimensional data. The advantage of using rough set theory is that it provides efficient and rigorous mathematical techniques for discovering patterns in data. However, it is only capable of handling discretized attribute values, but not the crisp and real-value attributes that neuro-fuzzy systems are capable of handling. The advantage of using the reject option in modeling classification data is that it improves the accuracy and reliability of the classification, but is not used in existing neuro-fuzzy systems.

Chapter 3 Rough set-based Neuro-Fuzzy System

3.1 Introduction

Neuro-fuzzy systems are popularly used in modeling data. They are gray-box models capable of abstracting humanly interpretable if-then fuzzy rules from available numerical data to perform universal approximation of non-linear systems. Their usage is qualitatively characterized by two features: interpretability and accuracy (Casillas et al., 2003). Hence, they can be classified as *linguistic neuro-fuzzy systems* that are focused on interpretability, mainly using the Mamdani model (Mamdani & Assilian, 1975); and *precise neuro-fuzzy systems* that are focused on accuracy, mainly using the Takagi-Sugeno-Kang model (Takagi & Sugeno, 1985).

Prevailing research on modeling data using linguistic neuro-fuzzy systems focused on increasing accuracy as much as possible but neglected interpretability (Casillas et al., 2003). Existing linguistic neuro-fuzzy systems employ the hybrid learning approach to abstract models from numerical data. In this approach, clustering is used in the first stage to generate the membership functions and competitive learning is used to identify the if-then fuzzy rules, followed by supervised learning that uses backpropagation in the final stage to optimize the membership functions. Although the unconstrained optimization in the final stage increases the accuracy of the abstracted model, it resulted in membership functions that are derailed from human-interpretable linguistic terms (de Oliveira, 1999b). Interpretability is vital to neuro-fuzzy systems in modeling data because if neglected, they degenerate into black-box models in which the advantages over other methods such as neural networks are lost (Casillas et al., 2003; Mikut et al., 2005).

In addition, a large number of if-then fuzzy rules are required to model high dimensional data, which in turn exceeds the human interpretation capacity (Tikk & Baranyi, 2003; Nauck, 2003). This interpretability issue on a large number of if-then rules motivates the complexity reduction of neuro-fuzzy systems (Kaynak, Jezernik, & Szeghegyi, 2002). This is similar to the problems encountered by numerical data driven techniques in data mining (Han & Kamber, 2001). These techniques rely on heuristics to guide or reduce their search space horizontally or vertically (Lin & Cercone, 1997). Horizontal reduction is realized by merging identical data tuples or the quantization of continuous numerical values while vertical reduction is realized by feature selection methods. In linguistic neuro-fuzzy systems, the former corresponds to the conversion of numerical inputs from a continuous range to a finite number of linguistic terms using membership functions while the latter corresponds to fuzzy if-then rule pruning and reduction. In some existing linguistic neuro-fuzzy systems, vertical reduction is employed by identifying fewer if-then fuzzy rules using certain heuristic thresholds such as the noise and ambiguity criteria (Quek & Zhou, 2001), or by applying pruning based on certainty factors (Chakraborty & Pal, 2004; Tung & Quek, 2002b). However, if the number of if-then rules is bounded as a practical limitation through the use of heuristic thresholds, then the universal approximation property is lost (Moser, 1999).

Recently, rough set theory (Pawlak, 1991) has been shown to significantly reduce pattern dimensionality and proven to be a viable data mining technique (Swinarski & Skowron, 2003). This motivates the synergy of the rough set method of knowledge reduction with neuro-fuzzy system in this work. A novel hybrid intelligent Rough set-based Neuro-Fuzzy System (RNFS) (Ang & Quek, 2006a) is proposed, with the objectives of modeling data with increased interpretability without compromising the

accuracy of the abstracted model, and of modeling high dimensional data without the use of heuristic thresholds in the identification or pruning of if-then fuzzy rules.

Details of the proposed RNFS are provided in the following sections. Section 3.2 describes the architecture of the RNFS. Section 3.3 explains the learning process of the RNFS, which includes the Supervised Pseudo-Self Evolving Cerebellar (SPSEC) algorithm that generates the membership functions and the Rough set-based Pseudo Outer-Product (RSPOP) algorithm that identifies the if-then fuzzy rules. The former is used if the membership functions are not manually defined, but the latter is essential in the modeling of all types of data. Section 3.4 describes the Mutual Information-based Rough Set Reduction (MIRSR) algorithm employed by RNFS to select features for modeling of high dimensional data. Section 3.5 describes the computation of Possibilistic Confidence Measure (PoCM) employed by RNFS to evaluate the reject option in modeling classification data. The relationships of these algorithms and the proposed RNFS are illustrated in Figure 3-1.

Figure 3-1: Relationships between the proposed algorithms and the RNFS

3.2 Architecture

The architecture of the Rough set-based Neuro-Fuzzy System (RNFS) is a five-layer neural network shown in Figure 3-2. Its architecture is developed using the Pseudo Outer-Product based Fuzzy Neural Network using Compositional Rule of Inference and Singleton fuzzifier (POPFNN-CRI(S)) (Ang, 1998; Ang et al., 2003) as a foundation. Hence, it resembles the architecture of the POPFNN shown in Figure 2-7.

Figure 3-2: Architecture of Rough set-based Neuro-Fuzzy System

The operations of each layer of the RNFS are the same as the POPFNN-CRI(S) described in section 2.4.4. Horizontal and vertical reduction in the RNFS are realized by merging identical data tuples using fuzzy sets and by reduction of redundant if-then fuzzy rules using rough set theory respectively. The if-then fuzzy rules of the RNFS are based on the Mamdani model given in equation (2.1). Although the architecture of RNFS resembles the five-layer neural network architecture from existing linguistic neuro-fuzzy systems such as FALCON (Lin & Lee, 1991), POPFNN (Zhou & Quek, 1996) and GenSoFNN (Tung & Quek, 2002b), the novel differences are:

- *If-then fuzzy rule reduction* – is performed in RNFS in which the RSPOP algorithm is used to identify if-then fuzzy rules, perform attribute reduction and rule reduction using rough set-based knowledge reduction. This vertical reduction process is performed autonomously without relying on user-defined heuristic thresholds to identify fewer if-then fuzzy rules or perform pruning.
- *Supervised learning* – is employed in RNFS in which the SPSEC algorithm is used to generate membership functions and the RSPOP algorithm is used to identify the if-then fuzzy rules, instead of using backpropagation or hybrid learning approach.
- *Feature selection based on wrapper approach* – is employed in RNFS to handle high dimensional data, in which the MIRSR algorithm is used to perform feature selection, instead of the filter approach.
- *Automated model abstraction* – is performed by RNFS without the need of user-defined parameters in the neural learning algorithms that generate the membership functions and identify the if-then fuzzy rules. Thus, its use does not require specialized skills and knowledge.

3.3 Supervised Learning

The RNFS employs a novel supervised learning approach that comprises mainly of two algorithms: the SPSEC that generates membership functions, and the RSPOP that identifies the if-then fuzzy rules. Learning algorithms in neuro-fuzzy literature are also broadly categorized into two kinds (Lin & Lee, 1996), namely, *parameter learning* that focuses on the updating of the weights of the network, and *structure learning* that is focuses on the change in the network structure. The SPSEC and RSPOP perform both parameter and structure learning simultaneously in the RNFS. The former operates on layers II and IV of RNFS whereas the latter operates on layer III of RNFS. They are described in the following sections in more details.

3.3.1 Generation of membership functions

The Supervised Pseudo Self-Evolving Cerebellar (SPSEC) algorithm is proposed to generate effective membership functions that reconcile with the semantics of interpretable linguistic variables. The SPSEC algorithm does not employ a further optimization process used in the hybrid learning approach to increase the accuracy of the abstracted model. This ensures that the interpretable membership functions generated do not deviate from human-interpretable terms. A description of the SPSEC membership generation algorithm is provided as follows with a detailed explanation on the rational of the algorithm.

3.3.1.1 Supervised Pseudo Self-Evolving Cerebellar (SPSEC) algorithm

- Step 1: Initialization
 - a. Given a dataset $\mathbf{X} = \{x_1, x_2, \dots, x_k, \dots, x_n\} \in \mathbb{R}$ of n samples, $\mathbf{Y} = \{\omega_1, \omega_2, \dots, \omega_k, \dots, \omega_n\} \in \{1, \dots, N_\omega\}$ where N_ω is the number of classes and n_ω data samples belonging to class $\omega \in \{1, \dots, N_\omega\}$, compute the standard deviation σ_ω of data samples that belongs to class ω . If the number of classes is unknown such that $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}$, create a default of $N_\omega=5$ output membership functions that span the universe of discourse of $\mathbf{Y}$ such that $\omega_k = l \mid \max_{l=1 \dots N_\omega} \mu_l(y_k)$ where the symbol $\mid$ means such that. The value of N_ω may also be specified by a user-defined value.
 - b. Set the initial number of neurons per class $m=200$.
 - c. Compute the neighborhood h_ω of class ω using normal optimal smoothing (Bowman & Azzalini, 1997) using equation

$$h_\omega = \left(\frac{4}{3n_\omega} \right)^{\frac{1}{5}} \sigma_\omega. \quad (3.1)$$

- Step 2: *Construct cerebellar structure*

Construct the initial cerebellar structure with m regularly spaced neurons $\mathbf{V}_\omega = \{v_1^\omega, v_2^\omega, \dots, v_m^\omega\}$ of class ω that span the range of $\left[\min_k x_k - \left(\left(\max_k x_k - \min_k x_k \right) / 4 \right), \max_k x_k + \left(\left(\max_k x_k - \min_k x_k \right) / 4 \right) \right]$ and initialize all neuron weights $\mathbf{W}_\omega = \{w_1^\omega, w_2^\omega, \dots, w_m^\omega\}$ to zero.

- Step 3: *Perform structural learning*

Perform a one-pass pseudo weight learning to discover the density distribution of the training data.

- Update the weight of the winning neuron where $v = x_k$ and the weights of neighboring neurons that belong to class $\omega = y_k$ using a Gaussian kernel

$$w_j^\omega = w_j^\omega + \frac{1}{n_\omega h_\omega \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{v_j^\omega - x_k}{h_\omega} \right)^2} \quad \forall j = 1..m, \forall k = 1..n, \quad (3.2)$$

where h_ω , computed in step 1c, defines the neighborhood width of influence of the winning neurons on neighboring neurons.

- Compute the normalized weights $\tilde{\mathbf{W}}_\omega = \{\tilde{w}_1^\omega, \tilde{w}_2^\omega, \dots, \tilde{w}_m^\omega\}$ across all classes using equation

$$\tilde{w}_j^\omega = \frac{w_j^\omega}{\sum_{\omega'=1}^{N_\omega} w_j^{\omega'}} \quad \forall \omega = 1..N_\omega, \forall j = 1..m. \quad (3.3)$$

- Step 4: *Evolve cerebellar structure*

Identify neurons with a high trophic factor, which are neurons with peak weights in a domain of the same class.

- Initialize peak of class ω $\mathbf{P} = \{p_1, p_2, \dots, p_j, \dots, p_m\}$ such that $p_j = 0 \quad \forall j = 1..m$.

- Initialize the domain $\mathbf{D} = \{d_1, d_2, \dots, d_j, \dots, d_m\}$ using equation

$$d_j = \omega \mid \tilde{w}_j^\omega = \max_{\omega'} \tilde{w}_j^{\omega'} \quad \forall j = 1..m. \quad (3.4)$$

- c. Compute the differenced neuron weights from neighbors using equation

$$\dot{w}_j^\omega = \begin{cases} \tilde{w}_j^\omega - 1/N_\omega & \text{if } j = 0 \\ \tilde{w}_{j+1}^\omega - \tilde{w}_j^\omega & \text{if } 0 < j < m \\ 1/N_\omega - \tilde{w}_{j-1}^\omega & \text{if } j = m+1 \end{cases} \quad \forall j = 1..m+1. \quad (3.5)$$

- d. Identify the peak weights from the differenced neuron weights using

$$p_j = \omega \text{ if } \left(\text{sgn}(\dot{w}_{j+1}^\omega) - \text{sgn}(\dot{w}_j^\omega) \right) < 0 \quad \forall \omega = 1..N_\omega, \forall j = 1..m. \quad (3.6)$$

- e. Insert peaks at both ends with $p_1 = d_1$ and $p_m = d_m$.

- f. Identify the highest peak in each domain of the same class that is representative of the neuron with a high trophic factor and remove lower peaks in each domain using equation

$$p_i = 0 \text{ if } \forall p_j, \exists p_i \mid \tilde{w}_i^\omega < \tilde{w}_j^\omega, p_j = p_i = d_i = \omega > 0 \quad \forall i, j \in [k, l], \quad (3.7)$$

where $[k, l]$ is the domain from $[1, m]$ such that $\forall j \in [k, l], d_j = \omega$.

- g. Adjust left peak towards the centre using equation

$$p_{j+1} = p_j, p_j = 0 \text{ if } \exists p_j \mid \tilde{w}_{j+1}^\omega = \tilde{w}_j^\omega, p_j = d_{j+1} = \omega, \forall j \in [1, k-1], \quad (3.8)$$

where $[1, k-1]$ is the left domain from $[1, m]$ such that $\forall j \in [1, k], d_j = \omega$.

- h. Adjust right peak towards the centre using equation

$$p_{j-1} = p_j, p_j = 0 \text{ if } \exists p_j \mid \tilde{w}_{j-1}^\omega = \tilde{w}_j^\omega, p_j = d_{j-1} = \omega, \forall j \in [k+1, m], \quad (3.9)$$

where $[k+1, m]$ is right domain from $[1, m]$ such that $\forall j \in [k, m], d_j = \omega$.

- Step 5: *Generate membership function*

Compute the parameters of the Gaussian fuzzy membership functions

$\mu_G(x; \alpha, \beta, \gamma, \beta)$ from neurons with a high trophic factor.

- a. Determine the number of membership functions J from the number of peaks $p_j > 0, j \in [1, m]$. Let the index of domain $\mathbf{D}$ be

$$[[l_1, r_1], [l_2, r_2], \dots, [l_J, r_J]] = [1, m] \quad \text{where} \quad d_i \neq d_j \quad \forall i \in [l_k, r_k],$$

$$j \in [l_{k+1}, r_{k+1}], \quad k \in [1, J-1].$$

- b. Compute the centre β_i of the i^{th} Gaussian membership functions using equation

$$\beta_i = v_j \mid p_j = d_j = \omega, j \in [l_i, r_i]. \quad (3.10)$$

- c. Compute the left parameter α_i using equation

$$\alpha_i = \begin{cases} \infty & \text{if } i = 1 \\ \frac{1}{\sqrt{2 \ln 2}} \left(\beta_i - \frac{v_{r_{i-1}} + v_{l_i}}{2} \right) & \text{if } 1 < i < J \end{cases}. \quad (3.11)$$

- d. Compute the right parameter γ_i using equation

$$\gamma_i = \begin{cases} \infty & \text{if } i = J \\ \frac{1}{\sqrt{2 \ln 2}} \left(\frac{v_{l_i} + v_{r_{i+1}}}{2} - \beta_i \right) & \text{if } 1 < i < J \end{cases}. \quad (3.12)$$

Figure 3-3 illustrates the neural learning process of the SPSEC algorithm whereby the intuition is to model the two-stage development process of the human nervous system. Figure 3-3(a) shows steps 1-2, where SPSEC constructs a cerebellar structure with m regularly spaced neurons that spans the input space. The cerebellum is the part of the human brain that involves the learning of motor skills and provides precise coordination of motor control for the human body parts (Kandel, Schwartz, & Jessell, 1995). The cerebellar cortex, with its regular and geometrical organization of different neurons (Lafarga, Berciano, Saurez, Andres, & Berciano, 1993), is modeled by these steps in the formation of the cerebellar structure. These steps also model the first-stage development process of the human nervous system where the basic architecture and coarse connection patterns are laid out without any activity-dependent processes (Kandel et al., 1995). Figure 3-3(b) shows steps 3-4, where SPSEC performs structural learning by performing a one-pass weight learning using Gaussian neighborhood learning to determine the distribution of the training data. SPSEC then pseudo self-evolves this cerebellar structure by identifying surviving neurons with a high trophic factor whose weights form a peak while the remaining neurons are

removed. These steps model the second-stage development process of the human nervous system where the initial architecture is refined in activity-dependent ways (Kandel et al., 1995). Figure 3-3(c) shows step 5, where the surviving neurons' weights are the parameters of the resulting Gaussian membership function that reconcile with the semantics of interpretable linguistic variables defined in section 2.4.1.

Figure 3-3: SPSEC neural learning process (a) Steps 1-2 constructs a basic cerebellar structure consisting of regularly spaced neurons, (b) Steps 3-4 active neurons stabilize through the update of trophic factors while competitors die, (c) Steps 5 surviving neurons form the resulting membership functions that reconcile with semantics of linguistic variables

3.3.2 Identification of if-then fuzzy rules

The Rough Set-based Pseudo Outer Product (RSPOP), which synergizes rough set knowledge reduction with the POP algorithm, is proposed to identify fewer if-then fuzzy rules than the POP algorithm so that the interpretability is improved without sacrificing the accuracy of the abstracted model. The proposed RSPOP algorithm consists of three parts: *RSPOP Rule Identification*, *RSPOP Attribute reduction* and *RSPOP Rule Reduction*. A description of the RSPOP algorithm is provided as follows with a detailed explanation on the rational of the algorithm.

3.3.2.1 Rough Set-based Pseudo Outer-Product (RSPOP) algorithm

RSPOP Rule Identification

- Step 1: *Initialization*

Initialize $\mathbf{C}_k^I$ and $w_{k,m,l}^{III}$ as in Step 1 of POP.

- Step 2: *Hebbian learning*

Given the training data tuples $\{\{\mathbf{X}_1, \mathbf{Y}_1\}, \{\mathbf{X}_2, \mathbf{Y}_2\}, \dots, \{\mathbf{X}_p, \mathbf{Y}_p\}, \dots, \{\mathbf{X}_n, \mathbf{Y}_n\}\}$, for $p=1..n$:

- Determine the input-labels $\mathbf{C}_{k,p}^I = \{c_1, c_2, \dots, c_i, \dots, c_{n_1}\}$ for the p^{th} instant of $\mathbf{X}$ with the maximum output for each input using equation

$$c_i = j_{\max} \mid \max_j(o_{i,j_{\max},p}^{II}) = \max_j(o_{i,j,p}^{II}) \forall i=1..n_1, c_i \in \mathbf{C}_{k,p}^I, \quad (3.13)$$

where k is the index to the rule node to be computed with $\mathbf{C}_{k,p}^I$ using (2.58); $o_{i,j,p}^{II}$ is the output of $IL_{i,j}^{II}$ for the p^{th} instant of $\mathbf{X}$; and $j_{\max}$ is the label index $j=1..J_i$ for a particular i^{th} input from $IL_{i,j}^{II}$ that yields the highest output for the p^{th} instant of $\mathbf{X}$.

- Compute the firing strength of the most influential rule node R_k^{III} using equation

$$f_{k,p}^{III} = \min_i (o_{i,c_i}^{II}), \quad (3.14)$$

where the firing strength $f_{k,p}^{III}$ is a measure of the contribution of rule node R_k^{III} to the fuzzy inference (Lin & Lee, 1996); o_{i,c_i}^{II} is the output of IL_{i,c_i}^{II} , $c_i \in \mathbf{C}_{k,p}^I$ for the p^{th} instant of $\mathbf{X}$.

- c. Compute the output-labels $\mathbf{D}_{k,p}^V = \{d_1, d_2, \dots, d_m, \dots, d_{n_5}\}$ for the p^{th} instant of $\mathbf{Y}$ with the maximum membership value using

$$d_m = l_{\max} \mid \mu_{m,l_{\max}}(y_{m,p}) = \max_l \mu_{m,l}(y_{m,p}) \forall m=1..n_5, d_m \in \mathbf{D}_{k,p}^V, \quad (3.15)$$

where $\mu_{m,l}$ is the membership function of $OL_{m,l}^{IV}$ for the p^{th} instant of $\mathbf{Y}$; $y_{m,p}$ is the m^{th} element of the p^{th} instant of $\mathbf{Y}$; and $l_{\max}$ is the label index $l=1..L_m$ for a particular m^{th} output from $OL_{m,l}^{IV}$ that yields the maximum membership value for the p^{th} instant of $\mathbf{Y}$.

- d. Update the weight w_{k,m,d_m}^{III} of the most influential rule node R_k^{III} using a Hebbian Learning rule (Hebb, 1949) as given in equation

$$\Delta w_{k,m,d_m}^{III} = f_{k,p}^{III} \times \mu_{m,d_m}(y_{m,p}) \forall m=1..n_5, d_m \in \mathbf{D}_{k,p}^V, \quad (3.16)$$

where w_{k,m,d_m}^{III} is the weight of R_k^{III} to OL_{m,d_m}^{IV} with the highest membership value; $f_{k,p}^{III}$ is the firing strength of the most influential rule node R_k^{III} for the p^{th} instant of $\mathbf{X}$ from (3.13)-(3.15); and μ_{m,d_m} is the maximum membership value of the m^{th} output for the p^{th} instant of $\mathbf{Y}$ from (3.15).

End for

- Step 3: *Rule formulation*

Determine the link of each rule node R_k^{III} to output-label nodes and update n_3 with the total number of rules as in Step 3 of POP.

RSPOP Attribute Reduction

- Step 1: *Baseline*

Given the training data tuples $\{\{\mathbf{X}_1, \mathbf{Y}_1\}, \{\mathbf{X}_2, \mathbf{Y}_2\}, \dots, \{\mathbf{X}_p, \mathbf{Y}_p\}, \dots, \{\mathbf{X}_n, \mathbf{Y}_n\}\}$ and the rules in POPFNN before reduction as $(\mathbf{C}^I, \mathbf{D}^V)$, compute the baseline objective measure of CD-rule base $(\mathbf{C}^I, \mathbf{D}^V)$.

For $p=1..n$:

Compute POPFNN consequence output for p^{th} instant of $\mathbf{X}$ using

$$CO_{m,l,p} = o_{m,l,p}^{IV} \quad \forall m=1..n_5, l=1..L_m. \quad (3.17)$$

End for

- Step 2: *Reduction*

Perform input attribute reduction for each attribute c_i of the rule base.

For $i=1..n_1$:

a. Compute the consistency of the CD-rule base $((\mathbf{C}^I - \{c_i\}), \mathbf{D}^V)$.

$((\mathbf{C}^I - \{c_i\}), \mathbf{D}^V)$ is inconsistent if equation (3.18) is true when rules with same conditions ignoring attribute c_i infer different consequents:

$$\begin{aligned} \forall R_k : ((\mathbf{C}_k^I - \{c_i\}), \mathbf{D}_k^V), \exists R_{k'} : ((\mathbf{C}_{k'}^I - \{c_i\}), \mathbf{D}_{k'}^V) | \\ ((\mathbf{C}_k^I - \{c_i\}) = (\mathbf{C}_{k'}^I - \{c_i\})) \wedge (\mathbf{D}_k^V \neq \mathbf{D}_{k'}^V) \quad \forall k, k' = 1..n_3. \end{aligned} \quad (3.18)$$

b. If $((\mathbf{C}^I - \{c_i\}), \mathbf{D}^V)$ is consistent, compute the objective measure of the CD-rule base $((\mathbf{C}^I - \{c_i\}), \mathbf{D}^V)$ using

$$CO_{m,l,p}^{\text{reduct}} = o_{m,l,p}^{IV} \quad \forall p=1..n, m=1..n_5, l=1..L_m. \quad (3.19)$$

c. Detect the deterioration of objective measure by comparing the objective measure computed in step 2b with the baseline objective measure. The objective measure is deteriorated if (3.20) is true when one of the correct consequent outputs $(CO_{m,l,p}, l = d_m)$ of using reduct is decreased. The object measure is also deteriorated if

the incorrect consequent output ($CO_{m,l,p}, l \neq d_m$) is greater than that of a correct consequent output:

$$\begin{aligned} & \exists p, \exists m \text{ s.t. } \forall l \left((CO_{m,l,p}^{reduct} < CO_{m,l,p}), l = d_m \in \mathbf{D}_{k,p}^V \right) \vee \\ & \left((CO_{m,l,p}^{reduct} > CO_{m,l,p}) \wedge (CO_{m,l,p}^{reduct} > CO_{m,d_m,p}), l \neq d_m \in \mathbf{D}_{k,p}^V \right) \quad (3.20) \\ & \forall p = 1..n, m = 1..n_5, l = 1..L_m. \end{aligned}$$

- d. If there is no deterioration of objective measure, then allow the reduction of attribute c_i using (3.21) and (3.22), remove duplicated rules and recompute n_3 :

$$(\mathbf{C}^I, \mathbf{D}^V) = ((\mathbf{C}^I - \{c_i\}), \mathbf{D}^V), \quad (3.21)$$

$$CO_{m,l,p} = CO_{m,l,p}^{reduct} \quad \forall p=1..n, m=1..n_5, l=1..L_m. \quad (3.22)$$

End for

RSPOP Rule Reduction

- Step 1: *Baseline*

Given the training data tuples $\{\{\mathbf{X}_1, \mathbf{Y}_1\}, \{\mathbf{X}_2, \mathbf{Y}_2\}, \dots, \{\mathbf{X}_p, \mathbf{Y}_p\}, \dots, \{\mathbf{X}_n, \mathbf{Y}_n\}\}$, and the rules in POPFNN before reduction as $(\mathbf{C}^I, \mathbf{D}^V)$, compute the baseline objective measure $CO_{m,l,p}$ of CD-rule base $(\mathbf{C}^I, \mathbf{D}^V)$ as in step 1 of RSPOP Attribute Reduction.

- Step 2: *Reduction*

Perform rule reduction for each attribute c_i of each R_k^{III} .

For $k=1..n_3, i=1..n_1$:

- a. Based on the picked rule $R_k^{III} : (\mathbf{C}_k^I, \mathbf{D}_k^V)$ and the picked condition attribute c_i , denoting CD-rule as $\phi \rightarrow \psi$, compute consistency of all CD-rules $\phi / (\mathbf{C}_k^I - c_i) \rightarrow \psi$ using equation

$$\begin{aligned} & \exists R_{k'} : ((\mathbf{C}_{k'}^I - \{c_i\}), \mathbf{D}_{k'}^V) | \\ & ((\mathbf{C}_k^I - \{c_i\}) = (\mathbf{C}_{k'}^I - \{c_i\})) \wedge (\mathbf{D}_k^V \neq \mathbf{D}_{k'}^V) \quad \forall k' = 1..n_3. \end{aligned} \quad (3.23)$$

- b. If the CD-rules $\phi / (C_k^I - c_i) \rightarrow \psi$ are consistent, remove attribute c_i from these CD-rules and compute the objective measure of POPFNN using these reduced rules using (3.19).
 - c. Detect deterioration of objective measures of using reduced rules if equation (3.20) is true.
 - d. If there is no deterioration of objective measures, allow the reduction of rules and remove duplicated rules.
- End for

The Pseudo Outer-Product (POP) algorithm, which is described in Section 2.4.4.2, is a simple, easy to understand one-pass learning algorithm that performs fast, reliable and efficient consequence identification of if-then fuzzy rules (Quek & Zhou, 2001). However, it has to compute all possible if-then fuzzy rules during the learning process in (2.55) (Quek & Zhou, 2001). Thus, a large number of if-then rules are usually identified. The total number of if-then fuzzy rules identified, $n_3 = \prod_{i=1}^{n_1} J_i$ as given in equation (2.43), also increases rapidly with an increase in the dimension of the input space or the number of input labels. The first part *RSPOP Rule Identification* addresses the inherent problem of the POP algorithm in considering all possible if-then fuzzy rules. In equation (2.55) of the POP algorithm, the weight $w_{k,m,l}^{III}$ is dependent on the firing strength $f_{k,p}^{III}$ for the p^{th} instant of training data $\{\mathbf{X}, \mathbf{Y}\}$. In the computation of $o_{m,l}^{IV}$, the highest $f_{k,p}^{III}$ from R_k^{III} results in the largest $o_{m,l}^{IV}$. This implies that the k^{th} rule with the highest firing strength $f_{k,p}^{III}$ is the most influential if-then fuzzy rule for the p^{th} instant of training data $\{\mathbf{X}, \mathbf{Y}\}$. However, it is not necessary to consider all possible if-then fuzzy rules to determine the rule index k to the rule node R_k^{III} with the highest $f_{k,p}^{III}$. The firing strength $f_{k,p}^{III}$ for the p^{th} instant of training data

$\{\mathbf{X}, \mathbf{Y}\}$ is computed from the same equation given in

$$f_{k,p}^{III} = \min_i (o_{i,j}^{II}) \quad (3.24)$$

where $o_{i,j}^{II}$ is the output of the input-label node $IL_{i,j}^{II}$ that forms the antecedent conditions for the i^{th} input to the k^{th} if-then fuzzy rule for the p^{th} instant of $\mathbf{X}$.

Expanding (3.24) with the objective of getting a maximum f_k^{III} gives

$$f_{k,p}^{III} = \min \left(\max(o_{1,c_1}^{II}), \max(o_{2,c_2}^{II}), \dots, \max(o_{i,c_i}^{II}), \dots, \max(o_{n_i,c_{n_i}}^{II}) \right), \quad (3.25)$$

where c_i is the label index $j=1..J_i$ for a particular i^{th} input from $IL_{i,j}^{II}$ that yields the highest output for the p^{th} instant of $\mathbf{X}$. Therefore, the maximum firing strength $f_{k,p}^{III}$ for the p^{th} instant of training data tuple $\{\mathbf{X}, \mathbf{Y}\}$ can be computed from the minimum of all the maximum outputs of the condition layer for each input. This allows the identification of the most influential rule node R_k^{III} with the conditions $\mathbf{C}_{k,p}^I = \{c_1, c_2, \dots, c_i, \dots, c_{n_i}\}$ where $c_i \in \mathbf{IL}_i$ for the p^{th} instant of training data tuple $\{\mathbf{X}, \mathbf{Y}\}$ in (3.13) of *RSPOP Rule Identification* step 2a.

The time complexity of the POP algorithm described in section 2.4.4.2 is $O(n_3)$, which is dependent on the number of if-then fuzzy rules. The time complexity of the reduction step 2a of both *RSPOP Attribute Reduction* and *RSPOP Rule Reduction* is $O(n_3^2)$ due to the two nested for loops involving k and k' . If POP is used, then these reduction steps have to consider a large number of if-then fuzzy rules of $n_3 = \prod_{i=1}^{n_i} J_i$ as given in equation (2.43). *RSPOP Rule Identification* improves POP such that given the p^{th} instant of training data tuple $\{\mathbf{X}, \mathbf{Y}\}$, RSPOP identifies only one most

influential if-then fuzzy rule. As this rule may have already been identified from the previous instant of the training data, the total number of if-then fuzzy rules identified using *RSPOP Rule Identification* is thus bounded from above as given in equation

$$n_3 \leq n, \quad (3.26)$$

where n is the number of training tuples; and n_3 is the number of if-then fuzzy rules identified. Thus, *RSPOP Rule Identification* improves the time complexity of POP by reducing the number of if-then fuzzy rules considered by POP, which is exponentially dependent on the dimension of the input space in equation (2.43), to an upper bound as given in equation (3.26). Comparing RSPOP against LazyPOP, the former identifies only one influential if-then fuzzy rule for a given training data tuple but the later identifies a number of legitimate if-then fuzzy rules based on a heuristic noise criterion. This gives RSPOP a stronger advantage in identifying significantly reduced number of if-then fuzzy rules for high dimensional modeling, which in turn improves interpretability of the if-then fuzzy rules identified.

To allow the reduction of attributes and rules in RNFS, the firing strength of the if-then fuzzy rules of the RNFS has to accommodate the removal of condition attributes. Therefore, the firing strength of a rule R_k^{III} in the rule layer of RNFS for handling reduced attributes and rules is modified to equation

$$f_k^{III} = \min_i \left(o_{i,c_i}^{II} \right), c_i \in C_k^I, c_i \neq \emptyset, \forall i = 1..n_1. \quad (3.27)$$

In the reduction of identified if-then fuzzy rules, the effectiveness of the reduced rule base must not deteriorate from rule base prior to the reduction. Therefore, an objective measure is vital in assessing the effectiveness of the RNFS before and after reduction. In rough set theory, there can only be one core of a knowledge, but there can be many

possible reducts (Pawlak, 1991). This objective measure thus serves the purpose of identifying the reducts that improve but not deteriorate the accuracy of the RNFS.

The fundamental objective of an if-then fuzzy rule identification algorithm is to identify if-then fuzzy rules that infer correct consequents based on the conditions and consequents learned from training data. Thus, an objective measure can be construed based on the correct inference of consequents from training data. Given the training data $\{\{\mathbf{X}_1, \mathbf{Y}_1\}, \{\mathbf{X}_2, \mathbf{Y}_2\}, \dots, \{\mathbf{X}_p, \mathbf{Y}_p\}, \dots, \{\mathbf{X}_n, \mathbf{Y}_n\}\}$, the l^{th} consequent of the m^{th} output for the p^{th} instant of training data $\{\mathbf{X}, \mathbf{Y}\}$ is $o_{m,l}^{IV}$ that is computed in the consequence layer. The correct consequent of the m^{th} output is computed from the maximum membership of output-labels $OL_{m,l}^{IV}$ given $y_{m,p}$. Therefore, an improvement in the objective measure is an increase in $o_{m,l_{\max}}^{IV} \mid \mu_{m,l_{\max}}(y_{m,p}) = \max_l \mu_{m,l}(y_{m,p})$. A deterioration in the objective measure is an increase in $o_{m,l}^{IV} \mid l \neq l_{\max}, o_{m,l}^{IV} > o_{m,l_{\max}}^{IV}$. This objective measure is used in *RSPOP attribute reduction* and *RSPOP rule reduction* to assess the performance of RNFS before and after reduction. The *RSPOP attribute reduction* and *RSPOP rule reduction* are responsible for performing vertical reduction in RNFS without relying on heuristic thresholds. They are applied from the same processes in rough set theory described in sections 2.5.2.3 and 2.5.2.4 respectively. A more detailed illustration of the operation of RSPOP in the process of identifying dispensable attributes and interpretable if-then fuzzy rules is given in section 4.3.

3.4 MI-based feature selection using wrapper approach

Existing neuro-fuzzy systems employ feature selection using filter approach to yield a smaller subset of features in order to handle large dimensional data. Prevailingly, Mutual Information (MI), which is a higher order statistics (Cover & Thomas, 2006),

is considered to be a good indicator of relevance between variables (Peng et al., 2005; Kwak & Chong-Ho Choi, 2002; Chow & Huang, 2005; Battiti, 1994). MI-based feature selection using filter approach is reviewed in Section 2.4.2.1. In the MI-based feature selection using the wrapper approach, the conditional entropy in (2.34) is simply

$$H(\Omega | \mathbf{X}) = - \sum_{\omega=1}^{N_{\omega}} P(\omega | \mathbf{X}) \log_2 P(\omega | \mathbf{X}), \quad (3.28)$$

where $P(\omega | \mathbf{X})$ is easy to estimate from the number of data samples classified as class ω using a modeling system as a classifier over the total number of data samples. An example of a wrapper approach, which uses the *Naïve Bayesian Parzen Window* (NBPW) classifier (Ang & Quek, 2006a), is similar to the MIPW algorithm described in section 2.4.2.2 but used (3.28) and the NBPW classifier instead of (2.38) to estimate the conditional entropy. This wrapper approach of performing feature selection using the NBPW classifier is called the *MI-based Naïve Bayesian Parzen Window* (MINBPW) algorithm in this work to differentiate it from the MIPW algorithm described in section 2.4.2.2.

Although feature selection using the wrapper approach is considered to yield better performance than the filter approach (Kohavi & John, 1997; Inza et al., 2002), feature selection algorithms using the wrapper approach are not available for linguistic neuro-fuzzy systems. Thus, the Mutual Information-based Rough Set Reduction (MIRSR) is proposed to perform feature selection using the wrapper approach in RNFS. A description of the MIRSR algorithm is provided as follows with a detailed explanation on the rational of the algorithm.

3.4.1 MI-based Rough Set Reduction (MIRSR) algorithm

- Step 1: *Generate membership functions*

Given n data $\mathbf{X}=\{X_1, X_2, \dots, X_i, \dots, X_d\}$ with d features, generate fuzzy membership functions of feature i using the SPSEC algorithm $\forall i=1..d$.

- Step 2: *Compute the MI of features $\mathbf{X}$ with the class Ω*

For $i=1..d$:

- Given the data $X_i=\{x_{i,1}, \dots, x_{i,k}, \dots, x_{i,n}\}$ in feature i , perform classification of each data $x_{i,k}$ using equation

$$\omega_{i,k} = p_{i,j} \mid \mu_{i,j} = \max_{j'=1..J_i} \mu_{i,j'}(x_{i,k}), \quad (3.29)$$

where $p_{i,j}$ is the class associated with the membership function $\mu_{i,j}$; and J_i is the number of membership functions generated for feature i .

- Estimate $P(\omega|X_i)$ from the number of correct classifications of class ω using X_i .
- Compute the MI of features $I(f_i; \Omega)$ using (2.33) where the conditional entropy $H(\Omega|X_i)$ of feature $f_i=\{X_i\}$ is estimated using equation (3.28).

End for

- Step 3: *Select the best k features*

Repeat

Select the feature f_i that maximizes $I(f_i; \Omega)$ using equation

$$F = F - \{f_i\}, S = S \cup \{f_i\} \mid I(f_i; \Omega) = \max_{j=1..d, f_j \in F} I(f_j; \Omega), \quad (3.30)$$

where F is the set of features to be selected from; and S is the set of features selected.

Until $|S|=k$

- Step 4: *Remove redundant features*

Remove membership functions that are not selected from step 3 and perform rough set reduction using RSPOP Attribute Reduction.

In step 3, the MIRSR algorithm relies on the specification of selecting an initial best k features so that the redundant features can be removed in step 4. An autonomous method is to select a portion of the total amount of features using equation (3.31).

$$k \approx 2 \log_2 d. \quad (3.31)$$

In feature selection literature, it is recognized that the selection of individually good features does not necessarily lead to good classification performance (Peng et al., 2005; Cover & Thomas, 2006; Cover, 1974). In other words, the simple *Best Individual Features* may not necessarily yield the best result. Nevertheless, it is useful to select some individually good features from a very large feature set so that feature selection can be performed by other methods that employ feature dependencies (Jain et al., 2000). Hence prevailing efforts to reduce the redundancy among features have resulted in a series of research (Ding & Peng, 2003; Peng et al., 2005; Jaeger, Sengupta, & Ruzzo, 2003). The rationale behind the proposed MIRSR algorithm is to first select some k individually good features from an overwhelmingly large feature set using MI where the conditional entropy is estimated from the membership functions of the RNFS, so that feature reduction of these k individually good features can be performed using *RSPOP attribute reduction*. The RSPOP algorithm is not capable of feature selection, but capable of performing feature reduction by employing the concept of knowledge reduction in rough set theory. Therefore, the proposed MIRSR algorithm synergizes the feature selection of a set of good features with high relevance using MI and the SPSEC algorithm with the feature reduction using the RSPOP algorithm to yield features with low redundancy.

Although the initial selection of k individually good features using MI seems to be a filter approach, the MI is estimated using the SPSEC algorithm employed in RNFS. In

contrast, feature selection algorithms based on the filter approach, such as MIBIFPW and MIPW, compute MI using Parzen Window to estimate the conditional entropy. These algorithms select features that are independent of the modeling system, but the proposed MIRSR algorithm selects features using the SPSEC and RSPOP algorithm that are employed in RNFS. Therefore, the proposed MIRSR is a feature selection algorithm based on the wrapper approach.

3.5 Computation of confidence measure

Confidence Measure (Kim & Ko, 2005) is a measure that reflects the classification reliability of predicting an unseen sample. This measure can be used to evaluate the option of rejecting the sample than risking a wrong classification, which can increase the accuracy of the system in modeling classification data (Yoma et al., 2005; De Stefano et al., 2000; Cordella et al., 1995). Based on psychological studies of human judgment as well as the probabilistic rejection and decision rule reviewed in section 2.6.1, a method of computing the Probabilistic Confidence Measure (PrCM) in modeling classification data is proposed in section 3.5.1. The PrCM is then extended to the Possibilistic Confidence Measure (PoCM) in section 3.5.2 using similar methodology that extends the probabilistic rejection and decision to the possibilistic approach in section 2.6.2.

3.5.1 Probabilistic Confidence Measure (PrCM)

In a psychological study on the estimation of probabilities of relatively unlikely events such as dying from cancer and smoking, respondents of several surveys have shown that people use the answer of ‘fifty-fifty’ as an expression for ‘I really don’t know’ (Fischhoff & Bruine de Bruin, 1999). The usage of ‘fifty-fifty’ corresponds to the assignment of equal probability to the two possible cases of something may or

may not happen where it is inferred from the ignorance of any ground for expecting one rather than the other (Dotterer, 1941). Given a pattern classification problem where the total number of classes $N_\omega=2$, and that the conditional probabilities of classes ω_1 and ω_2 given the data sample $\mathbf{x}$ are $p(\omega_1|\mathbf{x})=p(\omega_2|\mathbf{x})=0.5$; the confidence measure of the classification is intuitively at null. This is because the equal probability of the two classes corresponds to ignorance or no confidence in the classification. On the other hand, if the conditional probabilities of classes ω_1 and ω_2 given the data sample $\mathbf{x}$ are $p(\omega_1|\mathbf{x})=1$ and $p(\omega_2|\mathbf{x})=0$ or vice versa, the confidence measure of the classification is intuitively at unity. This is because the condition probability of one class at unity corresponds to a complete confidence in the classification.

Hence, a computation of the Probabilistic Confidence Measure (PrCM), which conforms to the boundary cases of no confidence and full confidence, can be interpolated using the difference of the maximum conditional probability of one of the classes with the second maximum probability over the total conditional probability of all the classes. The proposed computation of the PrCM is given in equation

$$CM(\mathbf{x}) = \frac{\max_{\omega_m \in \Omega} p(\omega_m | \mathbf{x}) - \max_{\omega_s \in \Omega'} p(\omega_s | \mathbf{x})}{\sum_{\omega \in \Omega} p(\omega | \mathbf{x})}, \quad (3.32)$$

where $p(\omega|\mathbf{x})$ is the conditional probability of class ω given the data sample $\mathbf{x}$; class $\omega \in \Omega$ in which $\Omega = \{1, \dots, N_\omega\}$ contains the set of all possible classes out of a total of N_ω classes; class $\omega_m \in \Omega$ is one of the classes that yield the maximum conditional probability; $\Omega' = \{\Omega - \omega_m\}$ is the set of all possible classes excluding class ω_m ; and class $\omega_s \in \Omega'$ is one of the classes that yield the second maximum conditional probability.

3.5.2 Possibilistic confidence measure (PoCM)

The method of computing the PrCM can be extended to Possibilistic Confidence Measure (PoCM) using the difference of the maximum conditional possibility of one of the classes with the second maximum possibility over the total conditional possibility of all the classes. The proposed computation of the PoCM is given in equation

$$CM(\mathbf{x}) = \frac{\max_{\omega_m \in \Omega} \pi_{\varpi|\chi}(\mathbf{x}, \omega_m) - \max_{\omega_s \in \Omega'} \pi_{\varpi|\chi}(\mathbf{x}, \omega_s)}{\sum_{\omega \in \Omega} \pi_{\varpi|\chi}(\mathbf{x}, \omega)}, \quad (3.33)$$

where $\pi_{\varpi|\chi}(\mathbf{x}, \omega)$ is the conditional possibility of $\varpi=\omega$ given $\chi=\mathbf{x}$; class $\omega \in \Omega$ in which $\Omega=\{1, \dots, N_\omega\}$ contains the set of all possible classes out of a total of N_ω classes; class $\omega_m \in \Omega$ is one of the classes that yield the maximum conditional possibility; $\Omega'=\{\Omega - \omega_m\}$ is the set of all possible classes excluding class ω_m ; and class $\omega_s \in \Omega'$ is one of the classes that yield the second maximum conditional possibility.

Using the computed CM in (3.32) or (3.33), the confidence-based rejection rule for the probabilistic or possibilistic approach is given by equation

$$\text{reject } \mathbf{x} \text{ if } CM(\mathbf{x}) < t, \quad (3.34)$$

where t is the rejection threshold, $0 \leq t \leq 1$. A more detailed explanation on the computation of an optimal rejection threshold for a specific classification task is discussed in section 6.2.

Although the method of computing the proposed PrCM and PoCM in equations (3.32) and (3.33) is not a unique method of interpolating the CM between the boundary cases, it addresses the issue of computing CM that reflects the classification reliability. The computation of the PoCM is employed in RNFS in the modeling

classification data, which helps to increase the accuracy of predicting unseen samples by evaluating the option of rejection the sample rather than risking a wrong classification and to indicate the reliability of the classification decision. A more detailed illustration on the estimation of classification confidence using the PoCM employed in RNFS is given in section 4.5.

3.6 Summary

This chapter proposes a novel hybrid intelligent Rough set-based Neuro-Fuzzy System (RNFS), which synergizes the sound concept of knowledge reduction from rough set theory with neuro-fuzzy systems. The RNFS is proposed with the objectives of modeling data with increased interpretability without compromising the accuracy of the abstracted model, and of modeling high dimensional data without the use of heuristic thresholds in the identification or pruning of if-then fuzzy rules.

The architecture of RNFS is a five-layer neural network developed using the Pseudo Outer-Product based Fuzzy Neural Network using Compositional Rule of Inference and Singleton fuzzifier (POPFNN-CRI(S)) as a foundation. The RNFS employs a novel supervised learning approach that comprises of the Supervised Pseudo Self-Evolving Cerebellar (SPSEC) algorithm and the Rough set-based Pseudo Outer-Product (RSPOP) algorithm. The proposed SPSEC addresses the problem of generating effective and interpretable membership functions by employing a two-stage learning technique that constructs a cerebellar-like structure whereby neurons with high trophic factors evolve to form the membership functions. The proposed RSPOP addresses the problem of generating effective and interpretable if-then fuzzy rules by synergizing rough set theory with the Hebbian learning technique to identify non-redundant if-then fuzzy rules without requiring user-defined parameters.

The RNFS also employs a novel Mutual Information-based Rough Set Reduction (MIRSR) algorithm that selects features using the wrapper approach and a novel method that computes the Possibilistic Confidence Measure (PoCM) of the RNFS. The proposed MIRSR addresses the problem of selecting features in modeling high dimensional classification data by synergizing the selection of relevant features and reduction of redundant features. The former is achieved by employing the MI computed using the membership functions generated from SPSEC and the latter is performed using RSPOP attribute reduction. The proposed PoCM addresses the problem of estimating the classification confidence of RNFS so that classification with low confidence can be rejected.

Chapter 4 Comparative Studies

4.1 Introduction

This chapter presents comparative studies of the proposed algorithms employed in the RNFS. Section 4.2 compares the generation of membership functions using the SPSEC algorithm proposed in section 3.3.1.1 against other membership function generation algorithms. Section 4.3 compares the abstracted model of using the RSPOP algorithm proposed in section 3.3.2.1 to identify the if-then fuzzy rules against the abstracted models using POP and other neuro-fuzzy systems. This section also gives a more detailed illustration on the attribute and rule reduction processes of the proposed RSPOP in the identification of interpretable if-then fuzzy rules. Section 4.4 compares the selection of features using the MIRSR algorithm proposed in section 3.4.1 against other MI-based feature selection algorithms. Finally, section 4.5 compares the estimation of classification confidence using the PrCM and PoCM proposed in sections 3.5.1 and 3.5.2 against the use of classifier output scores.

4.2 Generation of membership function

This section investigates the effectiveness of using the SPSEC algorithm proposed in section 3.3.1.1 in the generation of membership functions for modeling data. The effectiveness is investigated in terms of the interpretability of the membership functions generated and the capability of the membership functions in performing classification of the training data. The interpretability is assessed by considering the four semantic properties of interpretable linguistic variables reviewed in section 2.4.1; namely, *coverage*, *normalized*, *convex* and *ordered*. The capability of performing classification is assessed by considering the conformity of the decision boundaries of the membership functions generated to the inherent probabilistic decision boundaries of the training data.

The performance of the proposed SPSEC algorithm is compared against a selection of membership function generation algorithms; namely, FCM, LVQ and Bijective transform reviewed in sections 2.4.1.1, 2.4.1.2 and 2.4.1.3 respectively. Its performance is also compared against a sophisticated membership function known as the Discrete Incremental Clustering (DIC) (Tung & Quek, 2002a), which is employed to generate membership functions in GenSoFNN (Tung & Quek, 2002b).

4.2.1 Experiment: UCI Sonar dataset

A set of experiments is performed to investigate the effectiveness of the membership functions generated from the UCI Sonar dataset (Gorman & Sejnowski, 1988). This dataset is publicly available from (Newman, Hettich, Blake, & Merz, 1998), which consists of a total of $n=208$ patterns with $d=60$ features where 111 examples are sonar returns from metal and 97 are sonar returns from rock. The membership functions of the 52nd feature, arbitrarily chosen from the 60 features of the UCI Sonar dataset, are generated using two-thirds of the dataset as training data. This is due to the experimental process of performing 3-fold cross-validations on the entire UCI Sonar dataset in section 4.4.1. A more extensive experimental process involving both 3-fold and 10-fold cross-validations on larger datasets are presented in section 6.3.

Figure 4-1(a) shows the inherent probability distribution $p(x_{52}|\omega)$ estimated using Parzen Window and normal optimal smoothing reviewed in section 2.4.1.3 of the training data. Figure 4-1(b) shows the inherent posterior probability distribution $p(\omega|x_{52})$ of the training data computed from $p(x_{52}|\omega)$ using Bayes rule in equation (2.26), where the probabilistic decision boundaries capable of classifying the training data samples are highlighted.

Figure 4-1: Inherent distributions of x_{52} from the training data of the UCI Sonar dataset (a) probability distribution $p(x_{52}|\omega)$ estimated using Parzen Window and normal optimal smoothing; and (b) posterior probability $p(\omega|x_{52})$ computed from $p(x_{52}|\omega)$ using Bayes rule

Most clustering algorithms, such as FCM and LVQ, require the number of clusters c to be defined. The problem of finding an optimal c is known as *cluster validity* (Wang & Zhang, 2007), which is beyond the scope of this comparative study. In the following experiment using FCM and LVQ, a user-defined parameter of $c=3$ and 4 clusters are used. Any user-defined values of c can be used, but the same $c=3$ and 4 are used in this experiment for FCM and LVQ to illustrate the effect of c and to compare the results from these two algorithms.

Figure 4-2: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the FCM algorithm with parameters $c =$ (a) 3 clusters; and (b) 4 clusters

Figures 4-2(a) and (b) show the membership functions generated using FCM with $c=3$ and 4 clusters respectively. They are normalized, convex and ordered, but they do not cover the entire universe of discourse. In addition, the number of membership

functions generated is dictated by the parameter c , and the decision boundaries of the generated membership functions differ from the probabilistic decision boundaries in Figure 4-1(b).

Figures 4-3(a) and (b) show the membership functions generated using LVQ with $c=3$ and 4 clusters respectively. LVQ yielded only the centroids of the clusters. Thus, the widths of the membership functions generated are formed such that they are normalized, convex, ordered and cover the entire universe of discourse. However, the number of membership functions generated is dictated by the parameter c . In addition, the decision boundaries of the generated membership functions differ from the probabilistic decision boundaries in Figure 4-1(b).

Figure 4-3: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the LVQ algorithm with parameter $c =$ (a) 3 clusters; and (b) 4 clusters

The DIC algorithm requires two parameters to be defined; namely, *SLOPE* and *STEP*. Therefore, in the following experiment using DIC, user-defined parameters of $SLOPE=0.2$ and 0.3 , and $STEP=0.2$ and 0.4 are used. Any user-defined values of *SLOPE* and *STEP* can be used, but these two sets of values are used in this experiment to illustrate the effects of these two parameters.

Figures 4-4(a) to (d) show the membership functions generated using DIC with $\{SLOPE, STEP\} = \{0.2, 0.2\}$, $\{0.3, 0.2\}$, $\{0.2, 0.4\}$ and $\{0.3, 0.4\}$ respectively. They

are normalized, convex and ordered, but they do not cover the entire universe of discourse. Figures 4-4(a) and (c) show that the fuzziness or slope of all the membership functions generated using $SLOPE=0.2$ is identical. Similarly, Figures 4-4(b) and (d) show that the fuzziness or slope of all the membership functions generated using $SLOPE=0.3$ is identical. Instead of generating the fuzziness of the membership functions from the training data, DIC relies on $SLOPE$ to dictate the fuzziness of the membership function. Hence, instead of relying only on a user-defined parameter c in LVQ or FCM, the number of membership functions generated and their fuzziness are dictated by the parameters $SLOPE$ and $STEP$ in DIC. In addition, the decision boundaries of the generated membership functions differ from the probabilistic decision boundaries in Figure 4-1(b).

Figure 4-4: Membership functions of x_{s2} from the training data of the UCI Sonar dataset generated using the DIC algorithm with parameters $\{SLOPE, STEP\}$ = (a) $\{0.2, 0.2\}$; (b) $\{0.3, 0.2\}$; (c) $\{0.2, 0.4\}$; and (d) $\{0.3, 0.4\}$

Figure 4-5 shows the membership functions $\pi_{\omega|\chi}(x_{52}, \omega)$ autonomously generated using Bijective transform reviewed from the posterior probability distribution $p(\omega|x_{52})$ in Figure 4-1(b) without any user-defined parameters. They are normalized and they cover the entire universe of discourse, but they are not convex nor ordered. In addition, the decision boundaries of the generated membership functions differ from the probabilistic decision boundaries in Figure 4-1(b).

Figure 4-5: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the Bijective transform from the inherent posterior probability distribution

Figure 4-6 shows the membership function generated using the proposed SPSEC. They are normalized, convex, ordered and cover the entire universe of discourse. In addition, the decision boundaries of the generated membership functions in Figure 4-6 are similar to the probabilistic decision boundaries in Figure 4-1(b).

Figure 4-6: Membership functions of x_{52} from the training data of the UCI Sonar dataset generated using the proposed SPSEC algorithm

4.2.2 Discussion

Table 4-1 presents the consolidated experimental results of the generated membership functions of x_{52} using the various algorithms from the UCI Sonar dataset. The results show that LVQ is able to generate interpretable membership functions, which conform to all the semantic properties of interpretable linguistic variables. However, it relies on one parameter and its decision boundaries differ from the inherent probabilistic decision boundaries of the training data. Bijective transform, FCM and DIC do not generate interpretable membership functions and the decision boundaries of the generated membership functions do not conform to the inherent probabilistic decision boundaries of the training data, even though bijective transform does not require user-defined parameters. In contrast, the proposed SPSEC does not rely on any user-defined parameter, generates interpretable membership functions that conform to all the semantic properties of interpretable linguistic variables, and the decision boundaries of the membership functions generated conform to the inherent probabilistic decision boundaries of the training data.

Table 4-1 Consolidated results on the generation of membership functions of x_{52} from the training data of the UCI Sonar dataset

Membership function generation algorithms	Coverage	Normalized	Convex	Ordered	Number of user-defined parameters	Conformance to the probabilistic decision boundaries
FCM	✗	✓	✓	✓	1	✗
LVQ	✓	✓	✓	✓	1	✗
DIC	✗	✓	✓	✓	2	✗
Bijective transform	✓	✓	✗	✗	0	✗
Proposed SPSEC	✓	✓	✓	✓	0	✓

Therefore, the results show that the proposed SPSEC autonomously generates interpretable membership functions that are capable of performing classification without relying on any user-defined parameters.

4.3 Identification of if-then fuzzy rules

This section investigates the effectiveness of using the RSPOP algorithm proposed in section 3.3.2.1 in the identification of if-then fuzzy rules for modeling data. The effectiveness is investigated in terms of the interpretability and accuracy of the if-then fuzzy rules identified. The interpretability is assessed by the number of if-then fuzzy rules identified from the training data. The accuracy is assessed by the squared Pearson product-moment correlation value (R^2) and the mean square error (MSE) of the prediction on the test data.

The abstracted model in which the if-then fuzzy rules are identified using the proposed RSPOP is compared against the abstracted model using the POP reviewed in section 2.4.4.2. Both employ the same manually constructed membership functions that conform to the semantic properties of linguistic variables reviewed in section 2.4.1 without the use of any algorithms or heuristic parameters; and both are based on the same POPFNN-CRI(S) architecture reviewed in section 2.4.4. The performance of the abstracted model using the proposed RSPOP is also compared with other neuro-fuzzy systems; namely, Adaptive-Network-based Fuzzy Inference Systems (ANFIS) (Jang, 1993) with subtractive clustering (Chiu, 1994), Evolving Fuzzy Neural Networks (EFuNN) (Kasabov, 2001), and Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) (Kasabov & Song, 2002).

Two sets of experiments are performed to investigate the effectiveness of the if-then fuzzy rules identified from Nakanishi datasets in published paper (Nakanishi et al., 1993), and from real world highway traffic flow dataset collected by Tan (1997). The experiments are performed using Mathworks Matlab R2006a on a Dell Precision PWS380 PC equipped with Intel Pentium D 3.4 GHz and 2 GB of RAM.

4.3.1 Experiment: Nakanishi datasets

A set of experiments is performed to investigate the effectiveness of the if-then fuzzy rules identified from three published datasets; namely, a non-linear system, human operation of a chemical plant, and daily price of a stock in a stock market. These three datasets were used by Nakanishi, Turksen and Sugeno (1993) to evaluate six reasoning methods; namely, Mamdani version of Zadeh's CRI, Turksen's version of interval-valued CRI, Turksen's versions of point-valued and interval-valued AARS, Sugeno's version of position and gradient type of reasoning. They are also used in the classic paper of Sugeno and Yasukawa (1993) where data modeling based on fuzzy logic is proposed. Each of the three Nakanishi datasets is split into three groups: A, B and C in (Nakanishi et al., 1993). In this set of experiments, groups A and B are used as training data and group C is used as test data.

The abstracted model in which the if-then fuzzy rules are identified using the proposed RSPOP is compared against the abstracted model using POP. Both employ the same five manually constructed membership functions, and both are based on the same POPFNN-CRI(S) architecture. These two abstracted models are respectively abbreviated as RSPOP and POP in the experimental results. They are also compared against other abstracted models using ANFIS, EFuNN and DENFIS. ANFIS is configured with a range of influence of 0.5 for subtractive clustering. A smaller (or larger) selection on the range of influence for ANFIS will increase (or decrease) the number of if-then rules identified as well as the accuracy of ANFIS (Chiu, 1996). EFuNN and DENFIS are configured with default parameters.

4.3.1.1 Experiment 1: A nonlinear system

In this experiment, the if-then fuzzy rules are identified to model a nonlinear system. Figure 4-7 shows five Gaussian membership functions that are manually constructed

in this experiment to span the range of the inputs X_1 to X_4 and the output Y from the training data. The inputs and output are represented by five linguistic terms $\{ \text{very small, small, medium, large, very large} \}$ as labeled in Figure 4-7.

Figure 4-7: Five manually constructed membership functions for modeling a nonlinear system from Nakanishi datasets

Table 4-2 shows the initial and reduced if-then fuzzy rules identified using the proposed RSPOP and the constructed membership functions from the training data. The sets $\mathbf{C}^I = \{c_1, c_2, c_3, c_4\}$ and $\mathbf{D}^V = \{d_1\}$ are used to represent the condition and consequent attributes respectively. The attribute values $\{1, 2, 3, 4, 5\}$ are used to represent the 5 Gaussian membership functions for condition attributes c_1 to c_4 and consequent attribute d_1 .

Table 4-2: If-then fuzzy-rules identified using RSPOP for modeling a nonlinear system from Nakanishi datasets

Rules from RSPOP Rule Identification						Rules after RSPOP Attribute Reduction						Rules after RSPOP Rule Reduction					
$\mathbf{C}_k^I$					$\mathbf{D}_k^V$	$\mathbf{C}_k^I$					$\mathbf{D}_k^V$	$\mathbf{C}_k^I$					$\mathbf{D}_k^V$
k	c_1	c_2	c_3	c_4	d_1	k	c_1	c_2	c_3	c_4	d_1	k	c_1	c_2	c_3	c_4	d_1
1	3	4	2	1	1	1	3	0	2	1	1	1	3	0	2	1	1
2	2	1	3	1	3	2	2	0	3	1	3	2	2	0	3	1	3
3	1	2	4	1	3	3	1	0	4	1	3	3	1	0	4	1	3
4	5	3	5	1	1	4	5	0	5	1	1	4	5	0	5	0	1
5	3	4	1	2	1	5	3	0	1	2	1	5	3	0	1	2	1
6	1	1	2	2	5	6	1	0	2	2	5	6	1	0	2	2	5
7	4	1	2	2	2	7	4	0	2	2	2	7	4	0	0	2	2
8	2	1	3	2	5	8	2	0	3	2	5	8	2	0	3	2	5
9	4	2	5	2	2	9	4	0	5	2	2	9	0	0	2	3	2
10	2	2	2	3	2	10	2	0	2	3	2	10	3	0	3	3	4
11	4	2	2	3	2	11	4	0	2	3	2	11	2	0	4	3	2
12	3	1	3	3	4	12	3	0	3	3	4	12	1	0	1	4	2
13	2	2	4	3	2	13	2	0	4	3	2	13	5	0	1	4	1
14	5	4	5	3	1	14	5	0	5	3	1	14	4	0	2	4	2
15	1	4	1	4	2	15	1	0	1	4	2	15	1	0	3	4	4
16	5	4	1	4	1	16	5	0	1	4	1	16	4	0	3	0	1
17	4	2	2	4	2	17	4	0	2	4	2	17	2	0	4	4	3
18	1	2	3	4	4	18	1	0	3	4	4	18	3	0	5	4	2
19	4	5	3	4	1	19	4	0	3	4	1	19	2	0	1	5	2
20	2	1	4	4	3	20	2	0	4	4	3	20	1	0	2	5	3
21	3	1	5	4	2	21	3	0	5	4	2						
22	2	2	1	5	2	22	2	0	1	5	2						
23	1	3	2	5	3	23	1	0	2	5	3						
24	4	5	3	5	1	24	4	0	3	5	1						

Initially, the proposed RSPOP identified 24 if-then fuzzy rules before attribute and rule reduction. In the process of attribute reduction, one reduct $((\mathbf{C}^I - \{c_2\}), \mathbf{D}^V)$ that results in consistent rules is identified. Using this reduct, the computation of the objective measure resulted in no deterioration, and thus this reduct is used. After attribute reduction, the attribute c_2 is redundant, but the number of if-then fuzzy rules remained the same since there are no duplicated if-then fuzzy rules. In the process of rule reduction, several groups of rules are identified with dispensable conditions. An example of a pair of rules with a dispensable condition c_4 from the rules after RSPOP

attribute reduction is R_4 and R_{14} as illustrated in Table 4-2. The computation of the objective measure using this pair of reduced rules resulted in no deterioration. Thus, this pair of rules is reduced to R_4 after RSPOP rule reduction. As the rule reduction process continues, groups of reducible rules that resulted in no deterioration of objective measures are used with the dispensable conditions set to 0, and the number of if-then fuzzy rules is eventually reduced from 24 to 20.

The numbers 0-5 that represent the condition and consequent labels in Table 4-2 do not show that the if-then fuzzy rules are interpretable. To illustrate the intuitiveness of the if-then fuzzy rules identified, a mapping of the semantic labels to each condition and consequent labels is required. As reviewed in section 2.4.1, a linguistic variable is formally defined by Zadeh (1975) with a quintuple $(L, T(L), U, G, M)$. Here, the names L of the input and output linguistic variables are $\{X_1, X_2, X_3, X_4\}$ and $\{Y\}$, which are represented by condition and consequent variables $\{c_1, c_2, c_3, c_4\}$ and $\{d_1\}$, respectively. The linguistic terms are represented by the attribute values $\{1, 2, 3, 4, 5\}$. The meaning of each linguistic label M is represented by its Gaussian membership function in Figure 4-7. A mapping of semantic labels such as $T(\cdot) = \{ \text{very small}, \text{small}, \text{medium}, \text{large}, \text{very large} \}$ reveals the intuitiveness of the if-then fuzzy rules after RSPOP rule reduction in Table 4-2, as shown in Table 4-3.

After the if-then fuzzy rules are identified, the abstracted models are used to predict the test data of the nonlinear system. Figure 4-8 shows a plot of the predictions of the abstracted models using the proposed RSPOP and POP against the actual test data. The result shows that the predictions of POP are closer to the actual test data than the predictions of RSPOP. The accuracy of the predictions of the abstract models using POP, RSPOP and other neuro-fuzzy systems are compared using R^2 and MSE and

consolidated results are presented in Table 4-6 that includes two other experiments in sections 4.3.1.2 and 4.3.1.3. The analysis and discussion of the consolidated results are presented in section 4.3.1.4.

Table 4-3: Semantic interpretation of the if-then fuzzy rules in Table 4-2

R_1 :	IF X_1 is medium	AND X_3 is small	AND X_4 is very small
R_4 :	IF X_1 is very large	AND X_3 is very large	
R_5 :	IF X_1 is medium	AND X_3 is very small	AND X_4 is small
R_{13} :	IF X_1 is very large	AND X_3 is very small	AND X_4 is large
R_{16} :	IF X_1 is large	AND X_3 is medium	
	THEN Y is very small		
R_7 :	IF X_1 is large		AND X_4 is small
R_9 :	IF	X_3 is small	AND X_4 is medium
R_{11} :	IF X_1 is small	AND X_3 is large	AND X_4 is medium
R_{12} :	IF X_1 is very small	AND X_3 is very small	AND X_4 is large
R_{14} :	IF X_1 is large	AND X_3 is small	AND X_4 is large
R_{18} :	IF X_1 is medium	AND X_3 is very large	AND X_4 is large
R_{19} :	IF X_1 is small	AND X_3 is very small	AND X_4 is very large
	THEN Y is small		
R_2 :	IF X_1 is small	AND X_3 is medium	AND X_4 is very small
R_3 :	IF X_1 is very small	AND X_3 is large	AND X_4 is very small
R_{17} :	IF X_1 is small	AND X_3 is large	AND X_4 is large
R_{20} :	IF X_1 is very small	AND X_3 is small	AND X_4 is very large
	THEN Y is medium		
R_{10} :	IF X_1 is medium	AND X_3 is medium	AND X_4 is medium
R_{15} :	IF X_1 is very small	AND X_3 is medium	AND X_4 is large
	THEN Y is large		
R_6 :	IF X_1 is very small	AND X_3 is small	AND X_4 is small
R_8 :	IF X_1 is small	AND X_3 is medium	AND X_4 is small
	THEN Y is very large

Figure 4-8: Plot of the prediction results of the abstracted model using the proposed RSPOP and POP against actual test data of a nonlinear system from Nakanishi dataset

4.3.1.2 Experiment 2: Human operation of a chemical plant

In this experiment, the if-then fuzzy rules are identified to model the human operation of a chemical plant. Five Gaussian membership functions are manually constructed in this experiment to span the range of the inputs X_1 to X_5 and the output Y from the

training data. The illustration of these membership functions are omitted since it is similar to Figure 4-7. Table 4-4 shows the initial and reduced if-then fuzzy rules identified using the proposed RSPOP and the constructed membership functions from the training data. Initially, RSPOP identified 29 if-then fuzzy rules before attribute and rule reduction. In the process of attribute reduction, two reducts ($(C^I - \{c_1, c_2\}), D^V$) that results in consistent rules are identified. Using these reducts, the computation of the objective measure resulted in no deterioration. Thus, these reducts are used, and duplicated if-then fuzzy rules are removed. An example of duplicated if-then fuzzy rules using these reducts in Table 4-4 are R_6 and R_7 . After attribute reduction, the attributes c_1 and c_2 are redundant and the number of if-then fuzzy rules is reduced from 29 to 20 after removing duplicated rules. The number of if-then fuzzy rules is further reduced to 11 after rule reduction.

Table 4-4: If-then fuzzy rules identified using RSPOP for modeling the human operation of a chemical plant from Nakanishi datasets

Rules from RSPOP Rule Identification							Rules after RSPOP Attribute Reduction							Rules after RSPOP Rule Reduction						
C_k^I						D_k^V	C_k^I						D_k^V	C_k^I						D_k^V
k	c_1	c_2	c_3	c_4	c_5	d_1	k	c_1	c_2	c_3	c_4	c_5	d_1	k	c_1	c_2	c_3	c_4	c_5	d_1
1	1	4	5	2	1	5	1	0	0	5	2	1	5	1	0	0	5	0	1	5
2	3	4	2	3	1	2	2	0	0	2	3	1	2	2	0	0	2	3	0	2
3	2	4	5	3	1	5	3	0	0	5	3	1	5	3	0	0	0	4	1	4
4	2	2	3	4	1	4	4	0	0	3	4	1	4	4	0	0	3	5	1	3
5	1	4	4	4	1	4	5	0	0	4	4	1	4	5	0	0	3	0	2	3
6	1	3	5	4	1	5	6	0	0	5	4	1	5	6	0	0	5	0	0	5
7	2	4	5	4	1	5	7	0	0	3	5	1	3	7	0	0	3	0	0	3
8	3	3	3	5	1	3	8	0	0	2	3	2	2	8	0	0	4	2	4	4
9	4	3	3	5	1	3	9	0	0	3	3	2	3	9	0	0	1	0	4	1
10	3	1	2	3	2	2	10	0	0	5	3	2	5	10	0	0	5	0	0	5
11	3	2	3	3	2	3	11	0	0	3	4	2	3	11	0	0	4	3	5	4
12	1	4	5	3	2	5	12	0	0	5	4	2	5	12	0	0	5	4	2	5
13	2	1	3	4	2	3	13	0	0	3	1	4	3	13	0	0	3	1	4	3
14	1	3	5	4	2	5	14	0	0	3	2	4	3	14	0	0	3	2	4	3
15	1	4	5	4	2	5	15	0	0	4	2	4	4	15	0	0	4	2	4	4
16	2	4	5	4	2	5	16	0	0	1	3	4	1	16	0	0	1	3	4	1
17	3	4	3	1	4	3	17	0	0	5	3	4	5	17	0	0	5	3	4	5
18	4	5	3	2	4	3	18	0	0	1	4	4	1	18	0	0	1	4	4	1
19	1	2	4	2	4	4	19	0	0	5	4	4	5	19	0	0	5	4	4	5
20	5	2	1	3	4	1	20	0	0	4	3	5	4	20	0	0	4	3	5	4
21	5	4	1	3	4	1														
22	1	3	5	3	4	5														
23	1	4	5	3	4	5														
24	2	4	5	3	4	5														
25	4	2	1	4	4	1														
26	5	3	1	4	4	1														
27	1	3	5	4	4	5														
28	1	4	5	4	4	5														
29	1	3	4	3	5	4														

Figure 4-9 shows a plot of the predictions of the abstracted models using the proposed RSPOP and POP against the actual test data. The result shows that the predictions of RSPOP are closer to the actual test data than the predictions of POP. The accuracy of the predictions of the abstract models using POP, RSPOP and other neuro-fuzzy systems are compared using R^2 and MSE and consolidated results are presented in Table 4-6 that includes two other experiments in sections 4.3.1.1 and 4.3.1.3. The analysis and discussion of the consolidated results are presented in section 4.3.1.4. Similar to Table 4-3, a term set $T(\cdot)$ has to be used to map the attribute values of the if-then fuzzy rules in Table 4-4. The semantic interpretation of the if-then fuzzy rules is omitted henceforth, since translating them using a term set $T(\cdot)$ is a rather intuitive process illustrated in section 4.3.1.1.

Figure 4-9: Plot of the prediction results of the abstracted model using the proposed RSPOP and POP against actual test data of a chemical plant from Nakanishi datasets

4.3.1.3 Experiment 3: Daily price of a stock in a stock market

In this experiment, the if-then fuzzy rules are identified to model the daily price of a stock in a stock market. Five Gaussian membership functions are manually constructed to span the range of the inputs X_1 to X_{10} and the output Y from the training data. The illustration of these membership functions are again omitted since it is

similar to Figure 4-7. Table 4-5 shows the initial and reduced if-then fuzzy rules identified using the proposed RSPOP and the constructed membership functions from the training data. RSPOP identified an initial 50 if-then fuzzy rules before attribute and reduced to 28 after attribute and rule reduction.

Table 4-5: If-then fuzzy rules identified using RSPOP for modeling daily price of a stock in a stock market from Nakanishi datasets

Rules from RSPOP Rule Identification												Rules after RSPOP Rule Reduction											
k	C_k^I										D_k^I	k	C_k^I										D_k^I
	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}			c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	
1	4	2	3	2	2	1	3	3	1	1	3	1	0	0	0	2	0	0	3	3	1	1	3
2	4	2	3	2	2	3	2	1	2	2	2	2	0	0	0	2	0	0	2	1	2	2	2
3	4	1	3	1	1	3	1	3	3	2	4	3	0	0	0	1	0	0	1	3	3	2	4
4	4	3	3	3	3	2	3	3	2	3	2	4	0	0	0	3	0	0	3	3	2	3	2
5	2	2	2	2	3	3	2	3	3	3	3	5	0	0	0	2	0	0	0	3	3	3	3
6	3	3	3	3	3	3	2	3	3	3	5	6	0	0	0	3	0	0	2	3	3	3	5
7	3	2	2	2	4	4	2	3	3	3	3	7	0	0	0	2	0	0	2	4	3	3	3
8	1	2	1	2	3	3	3	3	3	3	3	8	0	0	0	2	0	0	5	2	4	3	3
9	2	2	2	2	4	4	3	3	3	3	3	9	0	0	0	2	0	0	3	3	4	0	3
10	3	2	3	2	4	3	2	4	3	3	3	10	0	0	0	2	0	0	3	0	3	4	3
11	4	2	4	2	2	5	5	2	4	3	3	11	0	0	0	2	0	0	2	3	0	4	3
12	1	2	2	2	3	3	3	3	4	3	3	12	0	0	0	1	0	0	2	4	3	4	3
13	4	2	4	2	2	5	3	2	3	4	3	13	0	0	0	2	0	0	4	4	0	4	3
⋮												14	0	0	0	0	0	0	4	4	3	4	1
37	5	3	5	2	4	3	4	4	4	4	3	15	0	0	0	3	0	0	3	3	4	4	4
38	2	2	2	2	4	4	4	4	4	4	3	16	0	0	0	2	0	0	0	3	4	4	3
39	4	5	4	5	5	4	4	4	4	4	1	17	0	0	0	2	0	0	2	4	4	4	2
40	4	2	4	2	4	4	2	2	5	4	3	18	0	0	0	5	0	0	0	4	4	4	1
41	1	2	1	2	4	4	3	3	5	4	3	19	0	0	0	3	0	0	2	4	4	4	3
42	2	2	1	2	4	4	3	4	5	4	3	20	0	0	0	2	0	0	3	4	0	4	3
43	4	4	3	4	4	4	3	4	4	5	2	21	0	0	0	2	0	0	2	2	5	4	3
44	5	4	4	4	5	4	4	4	4	5	2	22	0	0	0	2	0	0	3	3	5	4	3
45	5	4	5	4	5	4	4	4	4	5	2	23	0	0	0	4	0	0	0	4	4	5	2
46	4	4	4	4	5	3	3	5	4	5	2	24	0	0	0	4	0	0	3	5	4	5	2
47	4	5	3	5	3	4	5	4	5	1	1	25	0	0	0	5	0	0	4	5	4	5	1
48	5	3	5	2	4	3	3	4	5	5	2	26	0	0	0	2	0	0	3	4	5	5	2
49	5	4	5	4	5	4	3	4	5	5	1	27	0	0	0	4	0	0	3	4	5	5	1
50	2	2	2	3	4	3	3	5	5	5	2	28	0	0	0	3	0	0	3	5	5	5	2

Figure 4-10: Plot of the prediction results of the abstracted model using the proposed RSPOP and POP against actual test data of the daily price of a stock in a stock market from Nakanishi datasets

Figure 4-10 shows a plot of the predictions of the abstracted models using the proposed RSPOP and POP against the actual test data. The result shows that the predictions of POP are closer to the actual test data than the predictions of RSPOP. The accuracy of the predictions of the abstract models using POP, RSPOP and other neuro-fuzzy systems are compared using R^2 and MSE and consolidated results are presented in Table 4-6 that includes two other experiments in sections 4.3.1.1 and 4.3.1.2. The analysis and discussion of the consolidated result are presented in the next section.

4.3.1.4 Discussion

Table 4-6 shows the consolidated experimental results of experiments 1 to 3 from the Nakanishi datasets in terms of squared Pearson product-moment correlation value (R^2) and mean square error (MSE). Comparing the accuracy of predicting test data in terms of the lowest MSE and highest R^2 , ANFIS outperformed the rest on experiments 1 and 3, whereas DENFIS outperformed the rest on experiment 2. The results also show that ANFIS and DENFIS yielded good prediction accuracy on all three experiments. Comparing the averaged accuracy of predicting test data, precise neuro-fuzzy systems outperformed linguistic neuro-fuzzy systems. Since ANFIS and DENFIS are precise neuro-fuzzy systems based on the TSK model (Takagi & Sugeno, 1985), they generally yield increased accuracy but decreased interpretability (Casillas et al., 2003). In contrast, the abstracted models using the proposed RSPOP and POP are linguistic neuro-fuzzy systems based on the Mamdani model (Mamdani & Assilian, 1975), which is focused on interpretability. Comparing the accuracy of predicting test data, the proposed RSPOP and POP outperformed the average of precise neuro-fuzzy systems on experiments 2 and 3 but underperformed on experiment 1. The objective of comparing with precise neuro-fuzzy systems is to provide a quality check on the

performance of using the proposed RSPOP and POP in identifying if-then fuzzy rules since interpretability is not directly comparable between precise and linguistic neuro-fuzzy systems.

Table 4-6: Consolidated experimental results on the accuracy of abstracted models on three Nakanishi datasets

Experiment	Precise Neuro-Fuzzy Systems				Linguistic Neuro-Fuzzy Systems					
	Sugeno (P-G)	ANFIS	DENFIS	Average	Mamdani	Turksen (IVCRI)	EFuNN	POP	RSPOP	Average
1 Train MSE	N.A.	0.000	0.067		N.A.	N.A.	0.000	0.066	0.166	
Train R^2	N.A.	1.000	0.971		N.A.	N.A.	1.000	0.969	0.922	
Test MSE	0.467	0.286	0.411	0.388	0.862	0.706	0.566	0.482	1.304	0.784
Test R^2	0.845	0.853	0.805	0.834	0.490	0.609	0.720	0.735	0.362	0.583
2 Train MSE	N.A.	0.009x10 ⁴	0.445x10 ⁴		N.A.	N.A.	1.596x10 ⁴	8.975x10 ⁴	8.313x10 ⁴	
Train R^2	N.A.	1.000	1.000		N.A.	N.A.	0.999	0.993	0.991	
Test MSE	19.3x10 ⁵	30.1x10 ⁵	0.524x10 ⁵	16.7x10 ⁵	6.58x10 ⁵	2.58x10 ⁵	7.25x10 ⁵	3.72x10 ⁵	3.10x10 ⁵	4.65x10 ⁵
Test R^2	0.990	0.778	0.995	0.921	0.937	0.993	0.946	0.965	0.972	0.963
3 Train MSE	N.A.	0.0	21.1		N.A.	N.A.	0.9	30.4	34.5	
Train R^2	N.A.	1.000	0.933		N.A.	N.A.	0.997	0.912	0.904	
Test MSE	168.9	36.5	69.8	91.7	40.8	93.0	72.5	42.5	60.0	61.8
Test R^2	0.700	0.879	0.810	0.797	0.865	0.661	0.756	0.861	0.811	0.791

Table 4-7 shows a consolidated benchmark of the number of if-then rules identified and the prediction accuracy from the abstracted model using the proposed RSPOP against POP on the Nakanishi datasets.

Table 4-7: Consolidated benchmark of using POP against RSPOP in identifying if-then fuzzy rules on three Nakanishi datasets

Experiment	POP					RSPOP				
	Number of attributes	Number of training tuples	Number of rules identified	Train time (ms)	Test MSE	Number of rules identified initially	Train time (ms)	Number of rules after attribute reduction	Number of rules after rule reduction	Test MSE
1	4	25	625	93	0.482	24	213	24	20	1.304
2	5	35	3125	778	3.723×10^5	29	206	20	11	3.103×10^5
3	10	50	9765600	3907430	42.5	50	202	39	28	60.0

The results in Table 4-7 on interpretability show that as the dimensionality of experiments 1 to 3 increases from 4 to 10, the number of if-then fuzzy rules identified using the proposed RSPOP remains bounded from above by the number of tuples of the training data. In contrast, the number of if-then fuzzy rules identified using POP increases exponentially from 625 to over 9 million that is beyond human level interpretation. The results in Table 4-7 on the prediction accuracy in terms of MSE show that the proposed RSPOP outperformed POP in experiment 2, but

underperformed in experiments 1 and 3. A quality control check with other neuro-fuzzy systems shows that the proposed RSPOP outperformed ANFIS in experiment 2, but underperformed in experiments 1 and 3. It also outperformed DENFIS in experiment 3, but underperformed in experiments 1 and 2. These comparisons show that the prediction accuracy of the if-then fuzzy rules identified using the proposed RSPOP is not significantly compromised.

The results in Table 4-7 on computational complexity in experiment 3 shows that the training time utilized by the proposed RSPOP is 202 ms compared against 3,907,430 ms utilized by POP. This shows that the proposed RSPOP algorithm executes significantly faster than the POP algorithm. It is discussed in section 3.3.2.1 that the time complexity of the proposed RSPOP is $O(n_3^2)$ where n_3 in RSPOP is bounded from above by the number of training tuples. The time complexity of POP is $O(n_3)$ where n_3 in POP is exponentially dependent on the dimension of the input space given in equation (2.41) is also discussed. Since the number of if-then fuzzy rules identified by RSPOP is significantly smaller than POP as the dimensionality increases, the former thus executes significantly faster than the latter.

Therefore, the experimental results on Nakanishi datasets show that using the proposed RSPOP to identify if-then fuzzy rules increases the interpretability of the abstracted model without compromising the accuracy.

4.3.2 Experiment: Traffic flow prediction

A set of experiments is performed to investigate the effectiveness of the if-then fuzzy rules identified from the traffic flow dataset. The traffic flow dataset used is obtained from Tan (1997), which were collected at site 29 located at exit 15 along the eastbound Pan Island Expressway (PIE) in Singapore using loop detectors embedded

beneath the road surface. The inductive loop detectors were preinstalled by the Land Transport Authority of Singapore in 1996 along major roads to facilitate the collection of traffic flow data. Figure 4-11 shows a picture of the location where the traffic flow data were collected. There are five lanes at the site; namely, two exit lanes (lanes 4–5) and three straight lanes (lanes 1–3) for the main traffic.

Figure 4-11: Location of site 29 along the Pan Island Expressway in Singapore, where the traffic flow dataset (Tan, 1997) is collected

In this set of experiments, only the traffic flow data for the three straight lanes (lanes 1–3) are used. The traffic flow dataset consists of four attributes; namely, the time and the traffic densities of lanes 1–3. This traffic flow trend model is abstracted from this dataset, in which it is used to predict the traffic density for time $t + \tau$ where $\tau = 5, 15, 30, 45, 60$ minutes. Figure 4-12 shows a plot of the traffic flow density for lanes 1–3 spanning a period of six days from 5th to 10th September in 1996. The data set is divided into three cross-validation groups that comprise training and test sets exactly as in Tung and Quek (2002b) on the traffic flow prediction case study using GenSoFNN. These three data set groups are named CV1, CV2, and CV3 and are

extracted from the traffic flow data set as shown in Figure 4-12.

Figure 4-12: Plot of the traffic flow density of the three straight lanes along PIE at site 29

In the experiment, the abstracted model in which the if-then fuzzy rules are identified using the proposed RSPOP algorithm is compared against the abstracted model using POP. Both employ the same four manually constructed membership functions, and both are based on the same POPFNN-CRI(S) architecture. These two abstracted models are respectively abbreviated as RSPOP and POP in the experimental results. They are also compared against other abstracted models using ANFIS, EFuNN and DENFIS. EFuNN is configured with default parameters and DENFIS is configured with a distance threshold of 0.17. A smaller (or larger) selection on the distance threshold for DENFIS will increase (or decrease) the number of if-then rules identified as well as the accuracy of DENFIS (Kasabov & Song, 2002).

Figure 4-13 presents a plot of the recall on the training data and the prediction on the test data of the abstracted model using the proposed RSPOP to identify the if-then fuzzy rules. The MSE and the R^2 of prediction results are averaged to form the consolidated prediction results of lane 1 traffic flow density for $\tau = 5$ minutes.

Figure 4-13: Recall and prediction of lane 1 traffic density for $\tau = 5$ using the abstracted model where the if-then fuzzy rules are identified using the proposed RSPOP (a) recall for CV1; (b) prediction of CV2&3; (c) recall for CV2; (d) prediction of CV1&3; (e) recall for CV3; (f) prediction of CV1&2

Figure 4-14 presents 15 consolidated prediction results from using the various abstracted models in the prediction of lanes 1 to 3 for $\tau = 5, 15, 30, 45, 60$ minutes.

Figure 4-14: Consolidated experimental results on the accuracy of traffic flow prediction using the various abstracted models (a) average MSE of lane 1; (b) average R^2 of lane 1; (c) average MSE of lane 2; (d) average R^2 of lane 2; (e) average MSE of lane 3; (f) average R^2 of lane 3

The results in Figure 4-14 show that the abstracted model where the if-then fuzzy rules are identified using the proposed RSPOP yielded a lower average MSE error and higher R^2 when compared against the abstracted model using POP and other neuro-fuzzy systems.

4.3.2.1 Discussion

Table 4-8 shows a consolidated benchmark on the average number of if-then fuzzy

rules and the average MSE of the various abstracted models on modeling the traffic flow dataset. Comparing the accuracy of predicting test data in terms of the lowest MSE, RSPOP outperformed the rest. Comparing the interpretability of the abstracted model in terms of the lowest number of if-then fuzzy rules identified, DENFIS identified an average of 8.67 compared to an average of 24.16 by RSPOP. However, DENFIS is a precise neuro-fuzzy system based on the TSK model (Takagi & Sugeno, 1985), which generally yields decreased interpretability but increased accuracy (Casillas et al., 2003). In contrast, the abstracted models using the proposed RSPOP and POP are linguistic neuro-fuzzy systems based on the Mamdani model (Mamdani & Assilian, 1975), which is focused on interpretability. The objective of comparing with precise neuro-fuzzy systems is again to provide a quality check on the performance of using the proposed RSPOP and POP in identifying if-then fuzzy rules since interpretability is not directly comparable between precise and linguistic neuro-fuzzy systems.

Table 4-8: Consolidated benchmark on the average number of if-then fuzzy rules and average MSE of the various abstracted model on traffic flow prediction

Model	Number of if-then fuzzy rules	MSE
RSPOP	24.16	0.144
POP	256.00	0.163
GenSoFNN	50.00	0.164
EFuNN	234.11	0.189
DENFIS	8.67	0.155

The results show that the proposed RSPOP algorithm is able to identify fewer and effective if-then fuzzy rules, which improves the accuracy of the linguistic neuro-fuzzy system to the extent that it is comparable to precise neuro-fuzzy systems such as DENFIS. Hence, these results once again show that using the proposed RSPOP to identify if-then fuzzy rules increases the interpretability of the abstracted model without compromising the accuracy.

4.4 Feature selection for modeling high dimensional data

This section investigates the effectiveness using the MIRSR algorithm proposed in section 3.4.1 in the selection of features for modeling high dimensional data. The effectiveness of the proposed MIRSR algorithm using the wrapper approach is investigated in terms of the accuracy of RNFS in the classification of the test data set and the computation time utilized.

The performance of the proposed MIRSR algorithm is compared against a selection of MI-based feature selection algorithms and a rough set-based feature selection algorithm; namely, MIPW and MIBIFPW using the filter approach that are reviewed in sections 2.4.2.2 and 2.4.2.3 respectively, MINBPW for the NBPW classifier using the wrapper approach that is reviewed in section 3.4, and the New Fuzzy-Rough Feature Selection (FRFS2) algorithm (Jensen & Shen, 2007b).

4.4.1 Experiment: UCI Sonar dataset

A set of experiments is performed to investigate the effectiveness of the features selected from the same UCI Sonar dataset (Gorman & Sejnowski, 1988) used in section 4.2. The performance of RNFS on the UCI Sonar dataset is compared against the following classifiers:

- *Naïve Bayesian Parzen Window (NBPW)* – is a classifier that estimates $p(\mathbf{X}|\omega)$ and $P(\omega)$ from the training data samples and predicts the class ω with the highest posterior probability $p(\omega|\mathbf{X})$ using Bayes' rule in equation (2.26). The computation of $p(\omega|\mathbf{X})$ is rendered feasible by a naïve independence assumption that all the attributes $X_1, X_2, \dots, X_d$ are conditionally independent given class ω (Friedman, Geiger, & Goldszmidt, 1997) in equation

$$p(\mathbf{X}|\omega) = \prod_{i=1}^d p(X_i|\omega), \quad (4.1)$$

and by using Parzen Window in equation (2.22) to estimate the conditional probability $p(X_i|\omega)$.

- *Fisher Linear Discriminant (FLD)* – is a classifier that converts multi-dimensional data to one-dimension vector using a linear function that yields maximum ratio of between-class scatter to within-class scatter (Duda, Hart, & Stork, 2001).
- *Classification And Regression Tree (CART)* – is a classifier that uses symbolic tree-like representations of finite sets of if-then-else questions that are natural, intuitive and interpretable (Quinlan, 1986; Chou, 1991; Breiman, Friedman, Olshen, & Stone, 1984).
- *k-nearest neighbor (k-NN)* – is a classifier that assigns the class label of a new data based on the class with the most occurrences in a set of k nearest data points usually computed using a distance measure such as the Euclidean distance (Cover & Hart, 1967; Patrick & Fischer III, 1970; Dasarathy, 1991). A typical choice of k is a small odd number (1, 3 or 5) where a larger value reduces the effect of noise on the classification, but makes boundaries between classes less distinct. The $k=3$ -NN is used in this set of experiments.
- *Support Vector Machine (SVM)* – is a classifier that employs the kernel method, which maps the input space to a higher dimensional feature space so that linear classification of the feature space is equivalent to the non-linear classification of the input space (Vapnik, 1998; Schölkopf, Burges, & Smola, 1999). The SVM that employs the linear dot product kernel (Schölkopf et al., 1999) is used in this set of experiments.
- *Dynamic Evolving Neural-Fuzzy Inference System (DENFIS)* – a precise neuro-fuzzy system (Kasabov & Song, 2002), which was used in sections 4.3.1 and 4.3.2, is also used in this set of experiments.

The experiments are performed using Mathworks Matlab R2007a on a Dell Optiplex GX745 PC equipped with Intel Core2 Duo 2.4 Ghz and 2 Gb RAM. In this set of experiments, the training and test data sets are formed using 3-fold cross-validations. The experiments are repeated in three stages using different permutations of the training and test data sets and different classifiers using a selection feature selection algorithms.

There are two experimental approaches in performing feature selection using cross-validations; namely with and without selection bias. The former performs feature selection using the entire dataset where cross-validation accuracies are obtained from the smaller feature selected dataset. This approach performs feature selection only once, but introduces a selection bias that tends to yield results with relatively high cross-validation accuracies (Ambroise & McLachlan, 2002). The selection bias is present because cross-validation accuracies are obtained from the test data that are not external to the feature selection process. The latter performs both feature selection and training of the classifiers using the training data set at each stage of cross-validation. This approach corrects the selection bias because cross-validation accuracies are obtained from the test data that are external to the feature selection process (Ambroise & McLachlan, 2002). However, this approach performs feature selection multiple times, thus it is much lengthier to perform than the former approach. In this set of experiments, the latter approach is adopted in order to obtain cross-validation accuracies without selection bias.

In the experiments, MIBIFPW requires the specification of k best features to select. Thus, the experiments using MIBIFPW are repeated with an arbitrary selection of $k=3, 6, 9$ and 12 . On the other hand, MIPW and MINBPW employ equation (2.40) to stop the feature selection process and hence do not require the specification of k .

Since MINBPW and the proposed MIRSR are feature selection algorithms using the wrapper approach, the features selected by these two algorithms can only be used by the NBPW and RNFS classifiers respectively.

Table 4-9: Consolidated experimental results of the classification accuracies of using various MI-based feature selection algorithms and classifiers performed using 3-fold cross-validations without selection bias on the UCI Sonar dataset

Validation Method				3-fold cross-validation	
Feature Selection	Features	Exec (s)	Classifier	Training Acc(%)	Test Acc(%)
Filter Approach	MIBIFPW	k = 3 0.7 ± 0.0	NBPW	77.6 ± 3.3	77.4 ± 5.1
			FLD	72.8 ± 1.5	70.7 ± 5.4
			CART	91.3 ± 2.2	71.2 ± 6.2
			3-NN	84.6 ± 1.1	72.6 ± 4.4
			SVM	75.0 ± 2.2	74.5 ± 4.5
			RNFS	77.9 ± 2.1	75.5 ± 6.6
			DENFIS	79.8 ± 1.5	74.0 ± 2.9
		k = 6 0.7 ± 0.0	NBPW	77.6 ± 1.5	77.4 ± 4.5
			FLD	72.6 ± 2.7	66.4 ± 9.1
			CART	91.6 ± 1.5	74.5 ± 3.9
			3-NN	84.8 ± 3.4	76.9 ± 5.1
			SVM	75.7 ± 2.9	73.6 ± 5.1
			RNFS	83.9 ± 1.8	72.1 ± 5.2
			DENFIS	85.3 ± 4.8	73.5 ± 5.7
		k = 9 0.7 ± 0.0	NBPW	79.1 ± 1.5	75.0 ± 5.8
			FLD	70.9 ± 4.5	70.7 ± 6.1
			CART	93.5 ± 1.9	74.1 ± 2.7
			3-NN	85.6 ± 2.0	78.9 ± 7.0
			SVM	76.2 ± 3.2	75.0 ± 5.4
			RNFS	90.2 ± 2.3	70.2 ± 0.7
			DENFIS	91.1 ± 3.4	71.6 ± 6.0
	MIPW	k = 12 0.7 ± 0.0	NBPW	80.3 ± 1.8	74.5 ± 3.6
			FLD	69.7 ± 1.6	66.3 ± 4.4
			CART	93.5 ± 1.9	69.7 ± 1.5
			3-NN	84.4 ± 6.3	76.9 ± 6.2
			SVM	77.9 ± 1.5	75.5 ± 4.4
			RNFS	96.2 ± 1.1	72.6 ± 2.4
			DENFIS	94.2 ± 5.8	75.5 ± 6.8
		5.0 ± 0.0 6.5 ± 0.0	NBPW	80.8 ± 1.9	72.1 ± 7.5
			FLD	69.9 ± 3.6	67.3 ± 7.5
			CART	93.8 ± 1.5	75.0 ± 5.8
			3-NN	87.3 ± 3.6	77.9 ± 1.6
			SVM	73.8 ± 6.0	72.6 ± 2.7
			RNFS	84.1 ± 4.1	72.6 ± 2.9
			DENFIS	92.8 ± 4.4	75.5 ± 3.1
	FRFS2	5.0 ± 0.0 0.6 ± 0.0	NBPW	77.6 ± 2.2	74.0 ± 2.4
			FLD	72.3 ± 3.4	66.8 ± 4.1
			CART	91.6 ± 1.8	66.8 ± 2.9
			3-NN	90.1 ± 1.1	75.0 ± 3.2
			SVM	76.4 ± 1.6	69.2 ± 5.8
			RNFS	82.2 ± 5.0	72.6 ± 1.5
			DENFIS	93.0 ± 1.5	75.5 ± 4.1
Wrapper Approach	MINBPW	5.3 ± 1.2	1.0 ± 0.0	NBPW	85.6 ± 3.8
	MIRSR	12.0 ± 0.0	2.0 ± 0.2	RNFS	97.8 ± 1.3

Table 4-9 presents the consolidated experimental results of the training and test accuracies performed using 3-fold cross-validation without selection bias. Average

accuracies of using different classifiers on features selected using different MI-based feature selection algorithms are presented. The average number of features selected and the computation time taken by the various MI-based feature selection algorithms are also presented. The results show that RNFS that employed MIRSR using the wrapper approach yielded a superior test accuracy of $78.9 \pm 2.1\%$.

4.4.2 Discussion

Figure 4-15 shows a plot of the test classification accuracies against different classifiers using the selection of MI-based feature selection algorithms. The results also show that the RNFS using the proposed MIRSR yielded a superior test accuracy compared against NBPW that employed the MINBPW using the wrapper approach, and other classifiers that employed feature selection using the filter approach.

Figure 4-15: Consolidated results on the classification accuracies performed using 3-fold cross-validations without selection bias on the UCI Sonar dataset

The results in Figure 4-15 also show that the NBPW that employed MIBIFPW to select 3 best features yielded a superior test accuracy compared against the NBPW that employed MIBIFPW to select an increased number of best features or top-ranked features. This shows that selecting an increased number of best features may not

necessarily yield improved test accuracies. Comparing the number of features selected in Table 4-9, the proposed MIRSRS selected a larger number of features than MIPW, but yielded superior accuracy than MIBIFPW that selected a range of 3 to 12 features. This shows that the proposed MIRSRS is able to select more features with high relevance and low redundancy that improves the accuracy. Comparing the computational time taken in Table 4-9, the proposed MIRSRS utilized slightly longer time than MINBPW and FRFS2, but significantly shorter than MIPW.

Therefore, the results show that in modeling the UCI Sonar dataset using RNFS that employed the proposed MIRSRS using the wrapper approach, a superior accuracy is obtained without utilizing significantly large computation time, compared against other classifiers that employed feature selection using the filter approach.

4.5 Estimating classification confidence

This section investigates the estimation of classification confidence for modeling classification data. A set of experiments is performed to investigate the effectiveness of using the PrCM and PoCM proposed in sections 3.5.1 and 3.5.2 respectively, in the estimation of the classification confidence. The effectiveness is investigated by examining the uncertainty region at the decision boundary of the CM when the uncertainty in the data is increased. The proposed PrCM of NBPW and the proposed PoCM of RNFS in the estimation of classification confidence are compared against the use of the output scores of Support Vector Machine (SVM) (Vapnik, 1998) and Fisher Linear Discriminant (FLD) (Duda et al., 2001).

4.5.1 Experiment: Artificially generated dataset

It is known in psychology that if people were to make optimal decisions, they need to attach appropriate levels of uncertainty to types of beliefs (Harvey, 1997). Intuitively,

the confidence of making an optimal decision decreases with an increase in the level of uncertainty. Although uncertainty generally emerged from information deficiency (Klir, 1995), the three main types of uncertainty are recognized in the literature; namely, *nonspecificity*, *fuzziness*, and *strife* (Klir & Yuan, 1995). *Nonspecificity* results from imprecision in measurements, *fuzziness* results from vagueness in linguistic expression, and *strife* results from discord among various alternatives (Mendel, 2001; Klir & Yuan, 1995). For a recent review on a generalized theory of uncertainty, please refer to (Zadeh, 2006).

This investigation focuses on the nonspecificity type of uncertainty in the data as a result from the imprecision in measurements. In an example of a temperature measurement of 32.42 degrees with two significant numbers, a degree of uncertainty is included where the measurement is in the range of 32.41 to 32.43 degrees (Berthold, 2003). In fact, all measurements are intervals that has a width (or variance) that depends on the accuracy (or imprecision) of the measurements. In statistics, the Central Limit Theorem states that the distribution of the sample mean of a population with mean μ and standard deviation σ can be approximated with a normal distribution of mean μ and standard deviation $\sigma/\sqrt{n}$ when a large number of n samples are taken ($n \geq 30$) (Miller & Miller, 2004). Hence, a randomly generated dataset that comprises 200 data samples from a normal distribution where the standard deviation is increased is used to simulate an increased uncertainty in the data. The decision boundaries of the proposed PrCM from NBPW and the proposed PrCM from RNFS are compared against the decision boundaries of the output scores of the SVM and FLD classifiers when the uncertainty in the artificially generated data is increased.

Figure 4-16 shows the artificially generated dataset where 100 data samples belong to

class $\omega=1$ with a mean of $\mu_1=[0,0]$ and another 100 data samples belong to class $\omega=2$ with a mean $\mu_2=[1,1]$. Figures 4-16(a) and (b) show the dataset generated using a normal distribution with a standard deviation of $\sigma=0.2$ and $\sigma=0.3$ respectively. These two artificially generated datasets are similar to the first experimental dataset used by Li and Sethi (2006b) to demonstrate confidence-based classifier design using the output scores of an SVM classifier. The theoretical decision boundary of the artificially generated dataset is the line that intersects the points $[0,1]$ and $[1,0]$ as shown in Figure 4-16(a).

Figure 4-16: Plot of the generated dataset and support vectors where $\mu_1=[0,0]$ and $\mu_2=[1,1]$ with standard deviation of (a) $\sigma=0.2$ and (b) $\sigma=0.3$.

Figure 4-17: Surface plot of the output score of Support Vector Machine trained using the artificially generated dataset with (a) $\sigma=0.2$ and (b) $\sigma=0.3$.

Figures 4-17(a) and (b) show the surface plot of the output score of the SVM trained using the artificially generated dataset with $\sigma=0.2$ and $\sigma=0.3$ respectively. The output

scores of the trained SVM formed a plane in which the output score of a data sample is the normal distance of the data sample to the decision boundary such that it is positive for class $\omega=1$ and negative for class $\omega=2$.

The result in Figure 4-17(a) shows that the decision boundary of the SVM trained using the artificially generated dataset with $\sigma=0.2$ approximates the theoretical decision boundary. However, the result in Figure 4-17(b) shows that the decision boundary of the SVM trained using the artificially generated dataset with $\sigma=0.3$ differs significantly from the theoretical decision boundary. Thus, the results show that an increased in the uncertainty of the dataset from $\sigma=0.2$ to 0.3 significantly affects the decision boundary of the SVM.

Figures 4-18(a) and (b) show the surface plot of the output score of the FLD trained using the artificially generated dataset with $\sigma=0.2$ and $\sigma=0.3$ respectively. The output scores of the trained FLD formed a plane in which the output score of a data sample is the normal distance of the data sample to the decision boundary such that it is positive for class $\omega=1$ and negative for class $\omega=2$.

Figure 4-18: Surface plot of the output score of Fisher Linear Discriminant trained using the artificially generated dataset with (a) $\sigma=0.2$ and (b) $\sigma=0.3$.

The results in Figures 4-18(a) and (b) show that the decision boundaries of the FLD

trained using the artificially generated dataset with $\sigma=0.2$ and $\sigma=0.3$ correspond exactly with the theoretical decision boundary. Thus, the result shows that an increase in the uncertainty of the dataset from $\sigma=0.2$ to 0.3 does not affect the decision boundary of the FLD.

Figures 4-19(a) and (b) show the component probability distribution of the artificially generated dataset with $\sigma=0.2$ obtained using Parzen Window with univariate Gaussian kernel and normal optimal smoothing. Figures 4-19(c) and (d) show the component posterior probability distribution computed using Bayes rule given in equation (2.26). Figures 4-19(e) and (f) show the component probability distribution and Figures 4-19(g) and (h) show the component posterior probability distribution of the artificially generated dataset with $\sigma=0.3$. Comparing Figures 4-19(a) and (b) with Figures 4-19(e) and (f), the results show that the decision boundaries are similar, but an increase in the width of the component probability distributions is obtained when the uncertainty of the dataset is increased from $\sigma=0.2$ to 0.3. The use of Parzen Window is a non-parametric procedure that is capable of estimating arbitrary distributions without the assumption that the forms of the underlying densities are known (Duda et al., 2001). Thus, the use of Parzen Window yielded an increase in the width of the estimated component probability distributions, which corresponds to an increase in the spread or dispersion of the dataset when the standard deviation is increased from $\sigma=0.2$ to 0.3. Comparing Figures 4-19(c) and (d) with Figures 4-19(g) and (h), the results show an increase in the width of the uncertainty region when the uncertainty of the dataset is increased from $\sigma=0.2$ to 0.3. Since the component posterior probability distributions are computed from the component probability distributions, an increase in the latter results in an increase in the former.

Figure 4-19: Plot of the component probability distributions (a) $p(X_1|\omega)$, (b) $p(X_2|\omega)$, component posterior probability distribution (c) $p(\omega|X_1)$, (d) $p(\omega|X_2)$ of the artificially generated dataset with $\sigma=0.2$; plot of the component probability distribution (e) $p(X_1|\omega)$, (f) $p(X_2|\omega)$, component posterior probability distribution (g) $p(\omega|X_1)$, and (h) $p(\omega|X_2)$ of the artificially generated dataset with $\sigma=0.3$.

Figures 4-20(a) and (b) show the joint posterior probability distributions of the artificially generated dataset with $\sigma=0.2$ and $\sigma=0.3$ respectively. The joint probability distributions are computed from the component posterior probability distributions in Figures 4-19(c), (d), (g) and (h) using the Naïve Bayesian (NB) assumption employed in the NBPW classifier. Figures 4-20(c) and (d) show the proposed PrCM of Figures 4-20(a) and (b) respectively computed using equation (3.32).

Figure 4-20: Plot of the joint posterior probability distribution $p(X|\omega)$ of the artificially generated dataset with (a) $\sigma=0.2$ and (b) $\sigma=0.3$; and plot of the PrCM $CM(X)$ of the artificially generated dataset with (c) $\sigma=0.2$ and (d) $\sigma=0.3$.

The results in Figures 4-20(a) and (b) show that the decision boundaries of the NBPW trained using the artificially generated dataset with $\sigma=0.2$ and $\sigma=0.3$ are not straight lines but approximate the theoretical decision boundaries. Comparing Figures 4-20(a) and (b), the result shows an increase in the width of the uncertainty region at the decision boundary when the uncertainty of the dataset is increased from $\sigma=0.2$ to $\sigma=0.3$. The results in Figure 4-19 have shown that an increase in the uncertainty of the

dataset yielded an increase in the width of the component posterior probability distributions. Since the joint posterior probability distributions are computed from the posterior probability distributions, an increase in the latter results in an increase in the former. Comparing Figures 4-20(c) and (d), the results again show an increase in the width of the uncertainty region at the decision boundary of the PrCM of NBPW when the uncertainty of the dataset is increased from $\sigma=0.2$ to 0.3. Comparing Figures 4-20(c) and (d) with Figures 4-20(a) and (b), the results show that the PrCM $CM(X)=0$ at the decision boundary, but the joint posterior probability distribution $p(\omega_1|X)=p(\omega_2|X)=0.5$ at the decision boundary. Thus, the results show that PrCM corresponds intuitively to no confidence at the decision boundaries, but the joint posterior probability distribution does not have such intuitive correspondence.

Figure 4-21: Plot of the membership functions generated using the proposed SPSEC algorithm that characterize the possibility distribution (a) $\pi_{\omega|X}(X_1, \omega)$ and (b) $\pi_{\omega|X}(X_2, \omega)$ of the artificially generated dataset with $\sigma=0.2$; and plot of the membership functions generated using the proposed SPSEC algorithm that characterize the possibility distribution (c) $\pi_{\omega|X}(X_1, \omega)$ and (d) $\pi_{\omega|X}(X_2, \omega)$ of the artificially generated dataset with $\sigma=0.3$.

Figures 4-21(a) and (b) show the membership functions of the artificially generated dataset with $\sigma=0.2$ generated using the SPSEC algorithm proposed in section 3.3.1.1. Figures 4-21(c) and (d) show the membership functions of the artificially generated dataset with $\sigma=0.3$ similarly generated using the proposed SPSEC. Comparing Figures 4-21(a) and (b) with Figures 4-21(c) and (d), the results show that the decision boundaries are similar, but the width of the membership functions and the width of the uncertainty region at the decision boundary increases when the uncertainty of the dataset is increased from $\sigma=0.2$ to 0.3.

Figures 4-22(a) and (b) show the joint possibility distributions of the artificially generated dataset with $\sigma=0.2$ and $\sigma=0.3$ respectively. The joint possibility distributions are computed using equation (2.70) and the following if-then fuzzy rules employed in RNFS given in equation (4.2).

$$\begin{aligned} R_1: & \text{if } \chi_1 \text{ is } \textit{small} \text{ and } \chi_2 \text{ is } \textit{small} \text{ then } \varpi \text{ is } \omega_1 \\ R_2: & \text{if } \chi_1 \text{ is } \textit{large} \text{ and } \chi_2 \text{ is } \textit{large} \text{ then } \varpi \text{ is } \omega_2 \end{aligned} \quad (4.2)$$

where χ_i and ϖ are fuzzy linguistic variables that take values in the set X_i and Y respectively, and *small* and *large* are the membership functions in Figure 4-21.

Figures 4-22(c) and (d) show the proposed PoCM of Figures 4-22(a) and (b) respectively, computed using equation (3.33).

Figure 4-22: Plot of the joint possibility distribution $\pi_{\omega, \mathbf{x}}(\mathbf{x}, \omega)$ of the artificially generated dataset with (a) $\sigma=0.2$ and (b) $\sigma=0.3$; and plot of the PoCM $CM(\mathbf{x})$ of the artificially generated dataset with (c) $\sigma=0.2$ and (d) $\sigma=0.3$.

The results in Figures 4-22(a) and (b) show that the decision boundaries of the RNFS trained using the artificially generated dataset with $\sigma=0.2$ and $\sigma=0.3$ are not straight lines but approximate the theoretical decision boundaries. Comparing Figures 4-22(a) and (b), the result shows an increase in the width of the uncertainty region at the decision boundary when the uncertainty of the dataset is increased from $\sigma=0.2$ to 0.3 . Comparing Figures 4-22(c) and (d), the results again show an increase in the width of the uncertainty region at the decision boundary of PoCM of RNFS when the uncertainty of the dataset is increased from $\sigma=0.2$ to 0.3 . Comparing Figures 4-22(c) and (d) with Figures 4-22(a) and (b), the results show that the PoCM $CM(\mathbf{X})=0$ at the decision boundary, but the joint possibility distribution $\pi_{\omega, \mathbf{x}}(\mathbf{x}, \omega_1)=\pi_{\omega, \mathbf{x}}(\mathbf{x}, \omega_2)=0.5$ at the decision boundary. Thus, the proposed PoCM correspond intuitively to no

confidence at the decision boundaries, but the possibility distributions do not have such intuitive correspondence.

4.5.2 Discussion

The result in Figure 4-17 shows that SVM yielded approximately the theoretical decision boundary on the artificially generated dataset with $\sigma=0.2$. However, an increase in the uncertainty of the dataset to $\sigma=0.3$ affects the decision boundary of the SVM. The result in Figure 4-18 shows that the FLD yielded exactly the theoretical decision boundary on the artificially generated dataset with $\sigma=0.2$ and $\sigma=0.3$. This shows that an increase in the uncertainty of the dataset does not affect the decision boundary of the FLD. Although FLD is apparently more suitable than the SVM in the classification of the artificially generated dataset, the output scores of the FLD are the same although the uncertainty of the dataset is increased from $\sigma=0.2$ to 0.3. If the output score of the FLD is used to estimate the classification confidence, the output score should reflect a decrease in classification confidence when the uncertainty in the dataset is increased. Therefore, the results show that the output scores of the both SVM and FLD, which yielded the normal distance of a data sample to the decision boundary, are not suitable for estimating the classification confidence.

The results in Figures 4-20 and 4-22 show that the decision boundaries using the probability and possibility distributions of the NBPW and RNFS approximate the theoretical decision boundary. The widths of the uncertainty region at the decision boundaries increase when the uncertainty of the data is increased from $\sigma=0.2$ to 0.3, but the decision boundaries are not affected. The results also show that the proposed PrCM of NBPW and PoCM of RNFS are nil at the decision boundaries, but the probability and possibility distributions are half at the decision boundaries. Thus, the

proposed PrCM and PoCM correspond intuitively to no confidence at the decision boundaries, but the probability and possibility distributions do not have such intuitive correspondence.

Therefore, these results show that the proposed PrCM of NBPW and PoCM of RNFS are more suitable than using the output scores of the SVM or FLD for estimating the classification confidence.

4.6 Summary

This chapter demonstrates the effectiveness of the algorithms proposed in Chapter 3 that are employed in the RNFS. Section 4.2 presents experimental results on the membership functions generated using the proposed SPSEC algorithm versus a selection of membership generation algorithms from the UCI Sonar dataset. The results show that the proposed SPSEC is capable of generating membership functions that conform to the semantic properties of interpretable linguistic variables and inherent probabilistic decision boundaries of the training data without requiring user-defined parameters. Section 4.3 presents experimental results on the accuracy and number of if-then fuzzy rules identified using the proposed RSPOP algorithm versus a selection of neuro-fuzzy systems from the Nakanishi datasets and from a real world highway traffic flow dataset. The results show that the proposed RSPOP is capable of identifying a small number of humanly interpretable if-then fuzzy rules that increase the interpretability of the abstract model without compromising the accuracy. Section 4.4 presents experimental results on the classification accuracies of the RNFS that employed the proposed MIRS versus a selection of classification and feature selection algorithms from the UCI Sonar dataset. The results show that the RNFS that employed the proposed MIRS yields superior cross-validation accuracy. Finally,

section 4.5 presents the experimental results on the estimation of classification confidence using the proposed PoCM of RNFS versus the PrCM of NBPW and the output scores of SVM and FLD from artificially generated datasets. The results show that the proposed PoCM of RNFS and PrCM of NBPW are more suitable than the output scores of SVM or FLD for estimating the classification confidence.

Chapter 5 Financial Case Study: Stock Trading

5.1 Introduction

This chapter investigates the effectiveness of using the RNFS proposed in Chapter 3 in the modeling and forecasting of stock prices. The effectiveness is investigated in terms of the profits in the trading of stocks using the RNFS to forecast stock prices on real world stock market data. The trading profits measured using portfolio end values are considered rather than the statistical performance because this measure is important to an investor. The trading of stock using RNFS employs a forecast bottleneck free trading decision model whereby RNFS is used as the underlying predictor model and simple moving average trading rules are used to generate the trading decisions. The performance of stock trading with RNFS forecast model is compared against stock trading with the following: DENFIS forecast model, without forecast model and ideal forecast model.

5.2 Forecasting stock price

In forecasting stock prices from real world stock market, known stock prices have to be modeled as a time series in order to perform prediction on future stock prices. Thus, the forecasting of stock prices is an application of time series prediction (Box, Jenkins, & Reinsel, 1994). Time series prediction is concerned with the estimation of equations containing stochastic components, and thus can be modeled by an n^{th} -order linear difference equation with constant coefficients (Enders, 2004) in equation

$$y(T) = a_0 + \sum_{i=1}^n a_i y(T-i) + x(T), \quad (5.1)$$

where y is the time series where $y(T)$ represents a value at time instant T ; a_i is one of the arbitrary parameters that do not depend on any values of y or x ; and $x(T)$ is the term called the *forcing process* where it can be any function of time, current and

lagged values of other variables and/or stochastic disturbances.

If $x(T)$ is just a sequence of unspecified exogenous variables represented by a stochastic variable $\varepsilon(T)$ instead, then equation (5.1) resembles an autoregressive equation (Enders, 2004) and the time series prediction is formulated as: given values $y(T-n), y(T-n+1), \dots, y(T)$; predict $y(T+1)$ as $\hat{y}(T+1)$. This time series prediction using neural networks is known as time-delayed neural network in (Saad, Prokhorov, & Wunsch II, 1998); or recurrent neural network in (Pantazopoulos, Tsoukalas, Bourbakis, Brun, & Houstis, 1998). In the case of nonlinear time series prediction, it is known as the time delay embedding technique (Kugiumtzis, Lillekjendlie, & Christophersen, 1994) and n denotes the embedding dimension (Packard, Crutchfield, Farmer, & Shaw, 1980; Takens, 1981). The linear time series prediction in equation (5.1) is henceforth referred to as the *time-delayed approach* to discern it from recurrent networks with feedback connections. Several works applied this time-delayed approach using neural networks or neuro-fuzzy systems (Saad et al., 1998; Maguire, Roche, McGinnity, & McDaid, 1998; Pantazopoulos et al., 1998; Kimoto, Asakawa, Yoda, & Takeoka, 1990; Wang & Mendel, 1992). As a benchmark, the best estimation of equation (5.1) is the Random Walk model, which suggests that the day-to-day price change of a stock should have a mean value of zero (Enders, 2004). Formally, the Random Walk model asserts that the value of the time series should follow the stochastic difference equation (Enders, 2004) in equation

$$y(T) = y(T-1) + \varepsilon(T), \quad (5.2)$$

where $\varepsilon(T)$ is a normal random deviate with an expected mean of zero.

Therefore, the predicted value $\hat{y}(T)$ using the Random walk model is simply described by the equation

$$\hat{y}(T) = y(T-1). \quad (5.3)$$

Figure 5-1 illustrates the forecasting of stock price using the time-delayed approach. This approach employs n previous values of the series as input to the model and the predicted value as an output from the model. Each z^{-1} node represents a lag operation for one time instant. The model is trained using a series of known $y(T)$ prior to its use as a time series predictor. Specifically, assume that $y(1), y(2), \dots, y(N)$ are known, the training data set is formed using $N-n$ of these input-output data tuples.

Figure 5-1: Time-delayed price forecast approach adapted from (Ang & Quek, 2006b)

Prevailingly, a number of research publications of forecasting stock prices using neural networks based on the time-delayed price forecast approach did not compare against the Random Walk Model (see Appendix A in (Adya & Collopy, 1998)), except a recent work in (Chen, Leung, & Daouk, 2003). This is because the time-delayed approach performs direct forecast on a stock price series that is usually non-stationary. A time series is stationary, which is an important property in time series analysis, when the mean value of the series remains constant over the time series. Since many econometric time series are non-stationary, the time-delayed price forecast approach tends to include linear trends. Thus, deterministic shifts in the out-of-sample forecast period tend to exacerbate forecast errors (Clements & Hendry, 1999) when compared against the Random Walk Model.

Most non-stationary econometric time series can be transformed to stationary ones by using the method of differencing or cointegration (Clements & Hendry, 1999; Lin & Brannigan, 2003; Corradi & Swanson, 2006). Given values $y(T-n)$, $y(T-n+1), \dots, y(T)$, differencing these values give $\Delta y(T-n+1)$, $\Delta y(T-n+2)$, $\dots$, $\Delta y(T)$. Instead of predicting $y(T+1)$, predict the difference $\Delta \hat{y}(T+1)$ where Δ is the difference operator. Subtracting $y(T-1)$ from equation (5.1) gives rise to equation

$$\begin{aligned}\Delta y(T) &= a_0 + (a_1 - 1)y(T-1) + a_2 y(T-2) \dots + a_n y(T-n) + \varepsilon(T) \\ &= a_0 + (a_1 - 1)(y(T-1) - y(T-2)) + (a_2 + a_1 - 1)y(T-2) \dots \\ &\quad + a_n y(T-n) + \varepsilon(T) \\ &= a_0 + a'_1 \Delta y(T-1) + a'_2 \Delta y(T-2) \dots + a'_n \Delta y(T-n) + \varepsilon(T),\end{aligned}\quad (5.4)$$

where a'_i is another arbitrary parameter; $\varepsilon(T)$ is a stochastic random error term; and $\Delta y(T) = y(T) - y(T-1)$. Comparing equation (5.4) with (5.1), the former is a differenced version of the latter and $\hat{y}(T+1)$ can be obtained using equation

$$\hat{y}(T+1) = y(T) + \Delta \hat{y}(T+1) \quad (5.5)$$

This approach, which uses the difference operator Δ instead of the time-delayed price forecast approach in Figure 5-1, is known as the Time-delayed Price Difference Forecast approach (Ang & Quek, 2006b). This approach is illustrated in Figure 5-2.

Figure 5-2: Time-delayed price difference forecast approach adapted from (Ang & Quek, 2006b)

5.3 Experiment: Artificially generated stock price data

This section investigates the effectiveness of the time-delayed price difference forecast approach using RNFS proposed in Chapter 3 in the modeling of artificially generated stock price. The effectiveness is investigated in terms of the interpretability and accuracy of the model. The interpretability is assessed by the number of if-then fuzzy rules identified. The accuracy is assessed by statistical performance in terms of the squared Pearson product-moment correlation value (R^2) and the mean square error (MSE) on the accuracy of the prediction on the test data.

The performance of the time-delayed price difference forecast approach using RNFS is compared against the Random Walk model, and the other models using the same approach; namely, Feed-Forward Neural Network trained using Back-Propagation (FFNN-BP) (Rumelhart et al., 1986), Radial Basis Function Networks (RBFN) (Chen, Cowan, & Grant, 1991), Adaptive-Network-based Fuzzy Inference Systems (ANFIS) (Jang, 1993) with subtractive clustering (Chiu, 1994), Evolving Fuzzy Neural Networks (EFuNN) (Kasabov, 2001), and Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) (Kasabov & Song, 2002).

The artificial stock price data is generated using equations

$$y(T) = y(T-1) + \beta(T-1) + k\varepsilon(T), \quad (5.6)$$

$$\beta(T) = \alpha\beta(T-1) + v(T), \quad (5.7)$$

where α and k are constants; $\varepsilon(T)$ and $v(T)$ are normal random deviates with zero mean and unit variance. The artificially generated stock price data, which consists of 10000 data samples of which the first 3000 are used for training and the rest for testing, is shown in Figure 5-3. This dataset is generated similar to the simulation model used in (Moody, Wu, Liao, & Saffell, 1998) with the parameters $\alpha=0.9$, $k=3$

and initial conditions $y(0)=0$ and $\beta(0)=0$ using equations (5.6)-(5.7).

Figure 5-3: Artificially generated stock price data adapted from (Ang & Quek, 2006b)

In the experiments, FFNN-BP is configured with 10 input neurons, 30 hidden neurons and 1 output neuron using bipolar sigmoid activation function and is trained for 1000 training iterations using a learning rate of 0.025. RBFN is configured with a default spread of 0.1. ANFIS is configured with a range of influence of 0.3 for subtractive clustering and a maximum epoch of 100. EFuNN and DENFIS are configured with default parameters. RNFS is configured with two sets of different membership functions. The first configuration labeled as RNFS-1 employed manually constructed inputs and output membership functions using three and five fuzzy sets respectively. The inputs and output membership functions are constructed such that they span the range of $\pm|\Delta y(T)|$ as shown in Figures 5-4(a) and (b) respectively. The second configuration labeled as RNFS-2 employed inputs and output membership functions generated using SPSEC. Both the inputs and output membership functions generated are similar as shown in Figure 5-4(b). Both configurations employed RSPOP to identify the if-then fuzzy rules.

Figure 5-4: Membership functions (a) manually constructed for inputs $\Delta y(T)$ to $\Delta y(T-9)$ in RNFS-1; and (b) manually constructed for output $\Delta y(T+1)$ in RNFS-1 and generated for inputs $\Delta y(T)$ to $\Delta y(T-9)$ and output $\Delta y(T+1)$ using SPSEC in RNFS-2

Table 5-1 presents the experimental results of the recall on the training data and the prediction on the test data of the time-delayed price difference forecast approach using the aforementioned models.

Table 5-1: Experimental results of time-delayed price difference forecast approach using the selected models on artificially generated stock price data

Models	Neural Networks		Precise Neuro-Fuzzy Systems		Linguistic Neuro-Fuzzy Systems			Random Walk
	FFNN-BP	RBFN	ANFIS	DENFIS	EFuNN	RNFS-1	RNFS-2	
Train MSE ($\times 10^{-6}$)	16138	5.23	3.72	4.20	1.70	4.66	4.14	5.23
Train R^2	0.01	0.39	0.54	0.44	0.85	0.33	0.46	1.00
Test MSE ($\times 10^{-6}$)	50141	9.53	10.97	8.65	14.14	8.50	9.09	9.52
Test R^2	0.06	0.40	0.23	0.34	0.13	0.33	0.25	1.00
Number of if-then fuzzy rules	N.A.	N.A.	10	153	2167	80	1205	N.A.

Comparing the recall and predictive results in terms of lowest MSE; DENFIS, RNFS-1 and RNFS-2 yielded results that are superior to the Random Walk model. This shows that DENFIS and RNFS are superior choice than other models. The results also show that EFuNN with $\text{MSE}=1.70 \times 10^{-6}$ yielded the superior recall than other models, but its predictive results suffered. Moreover, RNFS-2 with $\text{MSE}=4.14 \times 10^{-6}$ yielded slightly better recall but poorer predictive results than RNFS-1. Therefore, both RNFS-2 and EFuNN demonstrated an over-trained phenomenon, which are evident in the large number of if-then fuzzy rules identified.

However, the results from just one artificially generated price data is not sufficiently convincing to conclude that DENFIS and RNFS models are superior choices compared against the other models. Therefore, further experimental results are collected from additionally generated artificial stock price data using equations (5.6)-(5.7). Table 5-2 presents the experimental results of the recall on training data and the prediction on test data of the time-delayed price difference forecast approach using the aforementioned models in terms of average MSE, average R^2 and average number of if-then fuzzy rules identified on a set of 20 artificially generated data.

Table 5-2: Averaged experimental results of time-delayed price difference forecast approach using selected models on 20 sets of artificially generated stock price data

Performance Measures	Neural Networks		Precise Neuro-Fuzzy Systems		Linguistic Neuro-Fuzzy Systems			Random Walk
	FFNN-BP	RBFN	ANFIS	DENFIS	EFuNN	RNFS-1	RNFS-2	
Train MSE ($\times 10^{-6}$)	15446 $\pm$ 3562	7.81 $\pm$ 5.52	5.28 $\pm$ 4.22	5.94 $\pm$ 4.12	2.48 $\pm$ 2.10	6.85 $\pm$ 4.84	5.88 $\pm$ 4.01	7.82 $\pm$ 5.53
Train R^2	0.01 $\pm$ 0.02	0.40 $\pm$ 0.02	0.58 $\pm$ 0.10	0.48 $\pm$ 0.02	0.85 $\pm$ 0.02	0.37 $\pm$ 0.04	0.52 $\pm$ 0.20	1.00 $\pm$ 0.00
Test MSE ($\times 10^{-6}$)	35894 $\pm$ 35137	9.69 $\pm$ 7.17	11.31 $\pm$ 8.61	8.60 $\pm$ 6.30	14.84 $\pm$ 10.15	8.81 $\pm$ 6.54	10.09 $\pm$ 7.19	9.68 $\pm$ 7.17
Test R^2	0.02 $\pm$ 0.06	0.41 $\pm$ 0.02	0.30 $\pm$ 0.08	0.37 $\pm$ 0.04	0.16 $\pm$ 0.03	0.32 $\pm$ 0.04	0.23 $\pm$ 0.09	1.00 $\pm$ 0.00
Number of if-then fuzzy rules	N.A.	N.A.	13.7 $\pm$ 13.1	191.6 $\pm$ 51.4	2138.2 $\pm$ 139.5	184.8 $\pm$ 64.4	2228.7 $\pm$ 873.6	N.A.

The results show that DENFIS and RNFS-1 yielded recall and predictive results that are superior to the Random Walk model. The results also show that EFuNN and RNFS-2 yielded better recall but poorer predictive results than the Random Walk model, demonstrating an over-trained phenomenon. The results also show that DENFIS yielded a slightly better predictive but inferior interpretability results in terms of MSE and of the number of if-then fuzzy rules respectively than RNFS-1. However, DENFIS is a precise neuro-fuzzy system based on the Takagi-Sugeno-Kang (TSK) model (Sugeno & Kang, 1988; Takagi & Sugeno, 1985), which generally yield decreased interpretability but increased accuracy (Casillas et al., 2003). In contrast, RNFS is a linguistic neuro-fuzzy system based on the Mamdani model (Mamdani & Assilian, 1975). The objective of comparing with a precise neuro-fuzzy system is to provide a quality check on the predictive performance of RNFS since interpretability

is not directly comparable.

This section presented the experimental results of the time-delayed price difference forecast approach using selected models on artificially generated sets of stock price data. The results show that the DENFIS and RNFS-1 yielded superior results than the Random walk model. The results also show that the RNFS-2 that employs SPSEC demonstrated over-trained phenomenon compared against RNFS-1. This shows that RNFS that employs SPSEC is able to model the training time-delayed price difference data effectively, but is not able to effectively predict the unseen price difference data. Hence, the time-delayed price difference forecast approach using RNFS that employs three and five manually constructed membership functions for the inputs and output respectively is used henceforth in this application.

A trading decision model and experimental results on the trading performance on stocks from real world stock market is presented in the next two sections to assess the trading performance of the time-delayed price difference forecast approach using DENFIS and RNFS.

5.4 Stock trading systems

In order to assess the trading performance of the stock trading with RNFS forecast model, a trading decision model where profits or losses are computable from the trading decisions is required. This section presents a realistic yet simple computation of the profits or losses of a trading decision model from (Moody et al., 1998). Two commonly used trading decision models based on the technical analysis approach from (Moody et al., 1998) are discussed; namely, with price forecast and without price forecast. A block diagram for a generic trading decision model without forecast based on the technical analysis approach is shown in Figure 5-5.

Figure 5-5: A trading decision model based on the technical analysis approach without forecast

The technical trading analysis in the model in Figure 5-5 is based on the premise that the market price patterns are assumed to recur in the future (Gencay, 1998). The price value of a stock is represented as a time series $y(T)$, where y represents a value at time instant T . The action of the trading system is assumed to be short, neutral or long represented by $F(T)$ where $F \in \{-1, 0, 1\}$ respectively (Moody et al., 1998). The trading profit of the trading system is subsequently modeled by the portfolio end value using a multiplicative return given in equation (5.8) where the transaction cost is assumed to be a fraction δ of the transacted price value (Moody et al., 1998).

$$R(T) = \{1 + F(T-1)r(T)\} \{1 - \delta |F(T) - F(T-1)|\} \quad (5.8)$$

where $r(T) = \frac{y(T)}{y(T-1)} - 1$; $F(T)$ is the trading decision from the trading system; and

δ is a fraction that represents the transaction cost.

A number of techniques can be used to generate buy and sell decisions using technical analysis techniques. One of the simplest and popular trading rules for deciding when to buy and sell in a security market is the moving average rule (Brock, Lakonishok, & LeBaron, 1992; Gencay, 1998). A widely used variant of the moving average rule is the Moving Average Convergence/Divergence (MACD) oscillator originally developed by Gerald Appel (Plummer & Ridley, 2003). MACD uses the crossovers of

a *Fast* signal given in equation (5.9) and a *Slow* signal given in equation (5.10) to indicate a buy or sell decision. The *Exponential Moving Average* (EMA) of a price series is given in equation (5.11) (Plummer & Ridley, 2003). The *Fast* signal in equation (5.9) is computed from the difference between the τ_{long} EMA and the τ_{short} EMA of $y(T)$ using the closing price where $\tau_{long} > \tau_{short}$. The *Slow* signal in equation (5.10) is computed from the τ_{slow} EMA of the *Fast* signal.

$$Fast(T) = EMA_{\tau_{short}}^y(T) - EMA_{\tau_{long}}^y(T), \quad (5.9)$$

$$Slow(T) = EMA_{\tau_{slow}}^{Fast}(T), \quad (5.10)$$

$$EMA_{\tau}^z(T) = Kz(T) + (1-K)EMA_{\tau}^z(T-1), \quad (5.11)$$

where $K = \frac{2}{\tau+1}$; τ is the number of time instants of the moving average; $y(T)$ is the price at the current time instant T ; and $EMA_{\tau}(T)$ is the τ EMA of time instant T .

Instead of using the crossover of the fast and slow signal to generate the buy/sell decision, simpler moving average trading rules can be generated using equation

$$F(T) = \text{sign}(Slow(T)), \quad (5.12)$$

where the slow signal $Slow(T)$ given in equation (5.10) is a moving average of the fast signal; the fast signal $Fast(T)$ is the two moving averages of the price level: a long-period average and a short-period average (Brock et al., 1992); and $F(T)$ is the trading decisions for time instant T .

Prevailing, the work on the modeling and forecasting of stock prices is motivated by the notion that the trading decision guided by forecasts are more profitable (Chen et al., 2003). A block diagram for a generic trading decision model based on the technical analysis approach with forecast is shown in Figure 5-6 (Moody et al., 1998).

Figure 5-6: A trading decision model based on the technical analysis approach with forecast adapted from (Moody et al., 1998)

In Figure 5-6, the forecasting model is trained using supervised learning on the training data and is then used to forecast unseen test data. The forecasts are then used as input to the trading system to generate the trading decisions. This common practice of using only the forecasts as input to the trading system results in a loss of information or a forecast bottleneck that may lead to suboptimal performance (Moody et al., 1998). An example on the use of this model is the work of (Chen et al., 2003) where trading decisions are generated using the degree of certainty on the directional forecast of price movements obtained from a trained Probabilistic Neural Network. Comparing the models in Figures 5-6 and 5-5, the former allows the generation of trading decisions from simple trading rules like the moving average trading rules while the latter only allows the generation of trading decisions based on the forecasts. This forecast bottleneck as indicated in (Moody et al., 1998) impedes the use of the price history of the input series $y(T)$ in the generation of trading decisions. As a result, the trading performance of the forecast module is encumbered. This motivates the research of a forecast bottleneck free trading decision model in Figure 5-7 (Ang & Quek, 2006b).

Figure 5-7: A forecast bottleneck free trading decision model with RNFS forecast adapted from (Ang & Quek, 2006b)

Comparing Figure 5-7 with Figure 5-6, the former model includes an additional signal of the input series $y(T)$ to the trading system to resolve the forecast bottleneck in the latter model, which appears to be an overly simplistic solution. Nevertheless, the former model facilitates the synergy of the time-delayed price difference forecast approach using RNFS in Figure 5-2 with the generation of trading decisions using simple moving average trading rules in equations (5.9)-(5.12). The computation of the trading decisions $F(T)$ using both the forecast value $\hat{y}(T+1)$ and $y(T)$ is given in equation (5.16). The computation of the moving average signals is given in the following equations:

$$Fast'(T+1) = EMA_{\tau_{short}}^{\hat{y}}(T+1) - EMA_{\tau_{long}}^{\hat{y}}(T+1), \quad (5.13)$$

$$Slow'(T+1) = EMA_{\tau_{slow}}^{Fast}(T+1), \quad (5.14)$$

$$EMA_{\tau}^{\hat{y}}(T+1) = K\hat{y}(T+1) + (1-K)EMA_{\tau}^y(T), \quad (5.15)$$

$$F(T) = \text{sign}(Slow'(T+1)), \quad (5.16)$$

where $K = \frac{2}{\tau+1}$; τ is the number of time instants of the moving average; $F(T)$ is the trading decisions from the trading system for time instant T ; and $\hat{y}(T+1)$ is the forecast price value from the forecast system.

5.5 Experiments: Trading of stock from real world stock market

This section investigates the effectiveness of trading stocks using RNFS to model and forecast stock prices from real world stock market. The effectiveness is investigated in terms of trading profits measured using portfolio end values $R(T)$ computed using equation (5.8). The performances of the stock trading with RNFS forecast model (labeled as *ST-RNFS*) are compared against stock trading with the following: DENFIS forecast model (labeled as *ST-DENFIS*), without forecast model (labeled as *ST-basic model*) and with ideal forecast (labeled *ST-ideal*) on 2 sets of stock price data; namely, Neptune Orient Lines (NOL) and Development Bank of Singapore (DBS). In these experiments, the trading decisions $F(T)$ for the ST-basic model is computed using the simple moving average trading rules in equations (5.9)-(5.12). The trading decisions $F(T)$ for the ST-DENFIS and ST-RNFS models, which employed the forecast bottleneck free trading decision model in Figure 5-7, are generated using the simple moving average trading rules in equations (5.13)-(5.16) where the forecast $\hat{y}(T+1)$ is obtained from the time-delayed price difference forecast approach using DENFIS and RNFS respectively. The trading decisions $F(T)$ for the ST-ideal model are computed using equations (5.13)-(5.16) where the actual price value $y(T+1)$ is used instead of $\hat{y}(T+1)$.

5.5.1 Neptune Orient Lines (NOL)

In this experiment, the daily stock prices of Neptune Orient Lines, which consists of 5917 price values, are obtained from Yahoo Finance web site on the counter N03.SI from the period of 2nd January 1980 to 1st March 2005. The training data comprises the first 4000 price values and the test data comprises the more recent 1917 price values. The forecasting models are trained with $n=10$ previous values of the differenced price data. Trading decisions are generated using equations (5.13)-(5.15)

with heuristically chosen parameters of $Long=12$, $Short=8$, and $Slow=5$. The portfolio end values $R(T)$ are computed using a transaction cost of $\delta=0.2\%$ that approximates the cost of online stock trading in Singapore. The moving average trading rule is modified by the introduction of a 1% moving average band. This hysteresis band is used to reduce the number of trading decisions by eliminating the “whiplash” signals when the short-period and long-period moving averages are close (Brock et al., 1992). Figure 5-8 shows a plot of the daily stock price, the long and short EMAs, and the trading decisions $F(T)$ of the ST-RNFS and ST-basic models on the test data of NOL.

Figure 5-8: Plot of price $y(T)$ and trading decisions $F(T)$ using the ST-RNFS and ST-basic models against time T on NOL

Figure 5-9: Plot of the portfolio end values $R(T)$ from the stock trading models against time T on NOL

Figure 5-9 shows the experimental results of the portfolio end values $R(T)$ obtained while trading stocks on the test data of NOL using the various stock trading models. All portfolio end values begin with a value of $R(T)=1.0$. The results shows that the ST-basic model yielded the lowest $R(T)=6.3027$ and the ST-ideal model yielded the highest $R(T)=73.2888$. The results also show that the ST-RNFS model yielded a superior $R(T)=16.5980$ than the ST-basic model and the ST-DENFIS model that yielded $R(T)=13.8675$.

Figure 5-10 shows the magnified results of Figure 5-8 from time $T=4500$ to $T=4580$. The moving average trading decisions are generated from the Slow EMA ($\tau_{\text{slow}}=5$) of the difference between the Long EMA ($\tau_{\text{long}}=12$) and Short EMA ($\tau_{\text{short}}=8$) of the test data of NOL. When the Short EMA crosses the Long EMA from below as indicated in Figure 5-10, it indicates a rising price trend and thus a buy decision ($F(T)=1$) is generated from the Slow EMA. The stock trading with forecast models generated the buy decision earlier with the forecast value $\hat{y}(T+1)$. Hence, the ST-RNFS and ST-DENFIS models yielded much higher portfolio end values than the ST-basic model. However, since the stock trading with forecast models are unable to forecast with exact accuracy, these models yielded a lower portfolio end value than the ST-ideal model. This result is intuitive because it is unlikely that returns higher than the ST-ideal model are obtainable since it uses the actual future price value $y(T+1)$ as the forecast value $\hat{y}(T+1)$.

Figure 5-10: Magnified results of Figure 5-8 from time $T=4500$ to $T=4580$

Figure 5-11 shows the manually constructed input and output membership functions, and the if-then fuzzy rules identified using the proposed RSPOP that are employed in the ST-RNFS model. RNFS employed 8 whereas DENFIS employed 9 if-then fuzzy rules. Since RNFS is a linguistic neuro-fuzzy whereas DENFIS is a precise neuro-fuzzy system, the if-then fuzzy rules in the former are more interpretable than the latter (Casillas et al., 2003). Moreover, Figure 5-11(c) shows that only 3 of the time-delayed price difference values ($\Delta y(T)$, $\Delta y(T-1)$, $\Delta y(T-2)$) are used in the stock trading with RNFS forecast model instead of the all $n=10$ time-delayed price difference values. Since the heuristically chosen value of $n=10$ does not represent the embedding dimension of the NOL price series, the results show that the RNFS computed an appropriate value of the embedding dimension. In contrast, the if-then fuzzy rules in the ST-DENFIS model used all the 10 time-delayed price difference values and the if-then fuzzy rules are too complex to be listed. Therefore, the results showed that the ST-RNFS model yielded a higher portfolio end value and more interpretable if-then fuzzy rules than the ST-DENFIS model.

Figure 5-11: (a) Manually constructed input membership functions employed in RNFS; (b) manually constructed output membership functions employed in RNFS; and (c) if-then fuzzy rules identified using RSPOP employed in RNFS on NOL

5.5.2 NOL with different trading parameters

The experiment in section 5.5.1 is repeated in this section using another set of heuristically chosen trading parameters where $Long=20$, $Short=10$, $Slow=4$ with the same 1% moving average band and the same transaction cost of $\delta=0.2\%$ to investigate the performance of the stock trading models using different trading parameters.

Figure 5-12: Plot of the portfolio end values $R(T)$ from the stock trading models against time T on NOL using another set of trading parameters

Figure 5-12 shows the experimental results of the portfolio end values $R(T)$ obtained while trading on the test data of NOL using the stock trading models with a different set of trading parameters, which begin with $R(T)=1.0$. The results show that the ST-basic model yielded the lowest $R(T)=9.2673$, and the ST-ideal model yielded the highest $R(T)=65.0778$. The results also show that the ST-RNFS model yielded a superior $R(T)=17.0362$ than the ST-basic model and the ST-DENFIS model that yielded $R(T)=15.2643$.

Comparing the results from this experiment using trading parameters $Long=20$, $Short=10$, $Slow=4$ against the results from the experiment using trading parameters $Long=12$, $Short=8$, $Slow=5$ in section 5.5.1, the results show that the use of different moving average parameters yielded different portfolio end values for all the stock trading models. The results also showed that the stock trading with forecast models yielded higher returns than the ST-basic model and yielded lower returns than the ST-ideal model. Most significantly, the results showed that the ST-RNFS model yielded a higher portfolio end value than the ST-basic model and the ST-DENFIS model, despite using a different set of moving average parameters.

5.5.3 Development Bank of Singapore (DBS)

In this experiment, the daily stock prices of Development Bank of Singapore (DBS), which consists of 6222 price values, are obtained from Yahoo Finance web site on the counter D05.SI from the period of 2nd January 1980 to 1st March 2005. The training data comprises the first 4000 price values and the test data comprises the more recent 2222 price values. The forecasting models are trained with $n=10$ previous values of the differenced price series as inputs. Trading decisions are generated using equations (5.13)-(5.15) with heuristically chosen parameters of $Long=12$, $Short=8$, $Slow=5$ with

a 1% moving average band and the portfolio end values $R(T)$ are computed using a transaction cost of $\delta=0.2\%$. The trading parameters and the transaction cost are the same as those used in section 5.5.1. Figure 5-13 shows a plot of the daily stock price, the long and short EMAs, and the trading decisions $F(T)$ of the ST-RNFS and ST-basic models on the test data of DBS.

Figure 5-13: Plot of price $y(T)$ and trading decisions $F(T)$ using the ST-RNFS and ST-basic models against time T on DBS

Figure 5-14: Plot of the portfolio end values $R(T)$ from the stock trading models against time T on DBS

Figure 5-14 shows the experimental results of the portfolio end values $R(T)$ obtained while trading on the test data of DBS using the stock trading models, which begin with $R(T)=1.0$. The results shows that the ST-basic model yielded the lowest $R(T)=2.0821$ and the ST-ideal model yielded the highest $R(T)=5.3635$. The results also show that the ST-RNFS model yielded a superior $R(T)=3.8582$ than the ST-basic model and the ST-DENFIS model that yielded $R(T)=2.4138$.

Figure 5-15: (a) Manually constructed input membership functions employed in RNFS; (b) manually constructed output membership functions employed in RNFS; and (c) if-then fuzzy rules identified using RSPOP employed in RNFS on DBS

Figure 5-15 shows the manually constructed input and output membership functions, and the if-then fuzzy rules identified using RSPOP that are employed in the ST-RNFS model. RNFS employed 5 whereas DENFIS employed 10 if-then fuzzy rules. Figure 5-15(c) shows that only 2 of the time-delayed price difference values ($\Delta y(T)$ and $\Delta y(T-1)$) are used in the ST-RNFS model instead of the all the $n=10$ time-delayed price difference value ($\Delta y(T)$, $\Delta y(T-1)$, ..., $\Delta y(T-9)$). Since the heuristically chosen value of $n=10$ does not represent the embedding dimension of the DBS price series, the results showed again that RNFS computed an appropriate value of the embedding

dimension. In contrast, the if-then fuzzy rules in the ST-DENFIS model used all 10 time-delayed price difference values and the if-then fuzzy rules are too complex to be listed.

This section presented the experimental results of trading stocks using RNFS in the modeling and forecasting of stock prices from real world stock markets. The results obtained on the NOL and DBS stock prices show that the ST-RNFS model yielded portfolio end values of $R(T)=16.598$ and 3.8582 respectively using the same moving average parameters $Long=12$, $Short=8$, $Slow=5$. These translate to profits of 15.598 and 2.8582 times the initial capital invested in NOL and DBS respectively. The results obtained on the NOL yielded a portfolio end values of $R(T)=17.0362$ that translates to a profit of 16.0362 using another set of moving average parameters $Long=20$, $Short=10$, $Slow=4$. Although a significant amount of profit is generated, the results once again exemplified the fact that profit or loss strongly depends on the trading module and its parameters, as well as the stock counter selected. These results showed a fair representation of the real world market because the effectiveness of the moving average trading rule relies heavily on the heuristic choice of moving average parameters for a particular data set. Hence, it is fair to generalize that profit or loss strongly depends on the trading module. Nevertheless, the results show that the ST-RNFS model yielded significantly higher trading profits than the ST-basic and ST-DENFIS models on stocks from real world stock market data. The results also showed that RNFS is capable of computing the embedding dimension of the stock price series, which resulted in a smaller number of interpretable if then fuzzy rules identified.

5.6 Summary

This chapter demonstrates the effectiveness of using the RNFS proposed in Chapter 3 in this financial case study of modeling and forecasting of stock prices. Experimental results on forecasting the price difference of artificially generated price series data showed that RNFS that employed manually constructed membership functions and DENFIS yielded superior predictive performance than the well-established Random Walk Model. However, the results showed that the RNFS that employed SPSEC to generate the membership functions demonstrated an over-trained phenomenon. This will be discussed in details in Chapter 7. As trading profits are more important to an investor than statistical performance, these two neuro-fuzzy systems are incorporated as the underlying predictor models with a forecast bottleneck free trading decision model (Ang & Quek, 2006b) where the trading decisions are generated using simple moving average trading rules.

Experimental results based on real world stock market data are presented on the stock prices of Neptune Orient Lines (NOL) and Development Bank of Singapore (DBS). Results on the trading profits in terms of portfolio end values of the stock trading with RNFS forecast (ST-RNFS) model are compared against the following stock trading models: with DENFIS forecast (ST-DENFIS), without forecast (ST-basic) and with ideal forecast (ST-ideal). Results showed that the ST-RNFS model identified fewer if-then fuzzy rules and yielded higher trading profits than the ST-DENFIS model. Therefore, the results showed that using RNFS as the underlying predictive model is able to identify if-then fuzzy rules with greater interpretability and accuracy than DENFIS. The results also provided evidence that profitability strongly depends on the trading module and its parameters and the stock counter selected. Despite the different yield in profits, the results consistently showed that the ST-RNFS model yielded

significantly higher profits than the ST-basic model and the ST-DENFIS model. Hence, the results showed that the ST-RNFS model is successful in assisting the trading module to increase its profitability from real world stock-trading applications.

The potential of the stock trading using RNFS model discussed in this chapter is encouraging as it demonstrates the feasible application of RNFS in the area of computational finance.

Chapter 6 Biomedical Case Study: Classifying cancerous samples

6.1 Introduction

This chapter investigates the effectiveness of using the RNFS proposed in Chapter 3 in the modeling and classification of biomedical data. The effectiveness is investigated in terms of the accuracy in classifying cancerous and normal tissues using RNFS that employs the MIRSR as proposed in section 3.4.1 to perform feature selection using wrapper approach and the PoCM as proposed in section 3.5.2 to reject classifications with low confidence. This chapter focuses on accuracy because the correct classification of cancerous from normal tissues and vice versa is imperative to medical doctors and the patients involved.

Two sets of experiments are performed to investigate the effectiveness of using RNFS in modeling biomedical data. In the first set of experiments, the performance of RNFS is compared against the NBPW that employs MINBPW using wrapper approach as described in section 2.4.2.2 to perform feature selection, and the PrCM as proposed in section 3.5.1 to reject classifications with low confidence on low dimensional biomedical data. Prevailingly, experimental techniques using DNA microarrays are capable of generating expressions of thousands of genes (Eisen & Brown, 1999) that are useful for differentiating cancerous from normal tissues (Alon et al., 1999). However, it is a major challenge to model this classification data because of the overwhelming gene expressions that are in the range of 2,000-30,000 whereby most of them are not useful to the classification problem (Wang, Makedon, Ford, & Pearlman, 2005). Hence, in the second set of experiments, the performance of RNFS that employs MIRSR on large dimensional microarray gene expression data is compared against a selection of classifiers and MI-based feature selection algorithms.

6.2 Rejecting classification with low confidence

This section investigates the effectiveness of using the PrCM and PoCM in rejecting classifications with low confidence on low dimensional biomedical data. The effectiveness of the proposed PoCM of RNFS employing the MIRSR feature selection algorithm using the wrapper approach is investigated in terms of the accuracy in the classification of the test data. The performance of the proposed PoCM of RNFS is compared against the published results in Li and Sethi (2006b), and the proposed PrCM of Naïve Bayesian Parzen Window (NBPW) employing the MINBPW feature selection algorithm using the wrapper approach.

6.2.1 Experiment: Pima Indian diabetes dataset

In this set of experiments, the Pima Indian diabetes dataset (Newman et al., 1998) is used, which consists of 768 tissue samples with 8 features where 268 samples are from patients with diabetes (class $\omega=2$) and 500 samples are from normal patients (class $\omega=1$). This dataset is used by Li and Sethi (2006b) to demonstrate the basic idea and advantages of a confidence-based classifier. The experimental setup is similar to Li and Sethi's (2006b) where the first 576 samples of the dataset are used for feature selection and training the model, and the remaining 192 samples are used for testing.

Table 6-1 shows the experimental results on the confusion matrix of using NBPW that employed MINBPW on the test data. The MINBPW selected 3 of the 8 features in the dataset, namely the 2nd feature on glucose, the 6th feature on Body Mass Index (BMI) and the 8th feature on age. As a benchmark, Li and Sethi's (2006b) experimental results yielded a training accuracy of 77.78% and a test accuracy of 77.08% using Support Vector Machine (SVM).

Table 6-1: Confusion matrix on the test data of the Pima Indians diabetes dataset using NBPW without rejection

Observation	Recall			Prediction		
	Negative	Positive	Total	Negative	Positive	Total
Negative	329	49	378	105	17	122
Positive	73	125	198	20	50	70

The results in Table 6-1 yielded a training accuracy of $(329+125)/(378+198)=78.82\%$ and a test accuracy of $(105+50)/(122+70)=80.73\%$. Therefore, the results of using NBPW that employed MINBPW yielded slightly better training and test accuracies as compared against Li and Sethi's (2006b) results using SVM. The results in Table 6-1 shows that among 70 patients with diabetes in the test data, 50 are classified correctly with diabetes and 20 are misclassified as normal. Therefore, the results show that the method of modeling biomedical data without rejecting the classifications with low confidence is not acceptable for medical diagnosis.

The joint posterior probability of a two-dimensional data can be visualized using a three-dimensional surface plot. However, in this experiment, MINBPW selected 3 features. Therefore, to visualize the joint posterior probability, Fisher Linear Discriminant (FLD) (Duda et al., 2001) is used to derive a transformation matrix so that the three-dimensional data is transformed to a two-dimensional plane that are efficient for discriminant (Duda et al., 2001). Figure 6-1 shows the surface plot of the joint posterior probability and the PrCM against FLD transformed features. Figures 6-1(a) and (b) show the joint posterior probability of class $\omega=1$ and that of class $\omega=2$ respectively. Figure 6-1(c) shows a superposition of Figures 6-1(a) and (b). Figure 6-1(d) shows the PrCM computed using equation (3.32).

Figure 6-1: Plot of the joint posterior probability distribution (a) $p(X|\omega_1)$ for class $\omega=1$; (b) $p(X|\omega_2)$ for class $\omega=2$; (c) $p(X|\omega)$ for both $\omega=1$ and 2 ; and (d) plot of the PrCM $CM(X)$ of NBPW against FLD transformed features t_1 and t_2 of the Pima Indians diabetes dataset

Figures 6-1(a) and (b) show that the joint posterior probability of a particular class is highest at the region where there are more training samples of the same class. Figure 6-1(d) shows that the samples from patients with diabetes ($\omega=2$) and normal patients ($\omega=1$) are not well separated, and a complex decision boundary is generated. The results show that there are a large number of samples from both classes at the complex decision boundary. Hence, the width of the uncertainty region at the complex decision boundary is significantly large. This result reinforces the studies presented in section 4.5 and shows that the proposed PrCM is suitable for estimating the classification confidence of the Pima Indian diabetes dataset so that classifications with low confidence can be rejected.

Next, the experiment is repeated using RNFS. Figure 6-2 shows the membership functions generated using the proposed SPSEC employed in RNFS. The MIRSRS employed in RNFS selected 3 out of the 8 features in the dataset; namely, the 2nd feature on glucose, the 6th feature on Body Mass Index (BMI) and the 8th feature on age. Table 6-2 shows the if-then fuzzy rules identified using the proposed RSPOP employed in RNFS.

Figure 6-2: Plot of the membership functions generated using the proposed SPSEC employed in RNFS that characterize the possibility distribution (a) $\pi_{mf|X}(X_2, \omega)$; (b) $\pi_{mf|X}(X_6, \omega)$; and (c) $\pi_{mf|X}(X_8, \omega)$ of the Pima Indians diabetes dataset

Table 6-2: If-then fuzzy rules identified using the proposed RSPOP employed in RNFS on the Pima Indians diabetes dataset

R_1	IF glucose is very large AND BMI is large AND age is small
R_{12}	IF glucose is small AND BMI is large AND age is medium
R_3	IF glucose is very large AND BMI is large AND age is medium
	THEN patient is diabetic
R_2	IF glucose is large AND BMI is medium AND age is small
R_4	IF glucose is large AND BMI is large AND age is small
R_5	IF glucose is large AND BMI is large AND age is medium
R_6	IF glucose is large AND BMI is medium AND age is medium
R_7	IF glucose is very large AND BMI is medium AND age is medium
R_8	IF glucose is very large AND BMI is medium AND age is small
R_9	IF BMI is medium AND age is large
R_{10}	IF glucose is very large AND BMI is large AND age is large
R_{11}	IF glucose is small AND age is small
R_{13}	IF glucose is large AND BMI is large AND age is large
	THEN patient is normal

The results show that the possibility distributions in Figure 6-2 resemble human interpretable linguistic terms and the if-then fuzzy classification rules in Table 6-2 are humanly interpretable. Therefore, the results show that RNFS yielded a model with higher interpretability than NBPW. Table 6-3 shows the experimental results on the confusion matrix of using RNFS that employed MIRSRS on the test data.

Table 6-3: Confusion matrix on the test data of the Pima Indians diabetes dataset using RNFS without rejection

Observation	Recall			Prediction		
	Negative	Positive	Total	Negative	Positive	Total
Negative	312	66	378	98	24	122
Positive	76	122	198	28	42	70

The results in Table 6-3 yielded a training accuracy of 75.34% and a test accuracy of 72.92%. The results of using RNFS that employed MIRSIR yielded slighter inferior training and test accuracies compared against NBPW that employed MIBBPW and Li and Sethi's results (2006b) using SVM. The results in Table 6-3 show that among 70 patients with diabetes in the test data, 28 are misclassified. Therefore, the results again show that the method of modeling biomedical data without rejecting the classifications with low confidence is not acceptable for medical diagnosis.

Figure 6-3: Plot of the joint possibility distribution (a) $\pi_{\omega\lambda}(X, \omega_1)$ for $\omega=1$, (b) $\pi_{\omega\lambda}(X, \omega_2)$ for $\omega=2$, (c) $\pi_{\omega\lambda}(X, \omega)$ for both $\omega=1$ and 2 and (d) plot of the PoCM $CM(X)$ of RNFS against transformed FLD features t_1 and t_2 of the Pima Indians diabetes dataset

Figure 6-3 shows the surface plot of the joint possibility and the PoCM against FLD transformed features. Figures 6-3(a) and (b) show that the joint possibility of class $\omega=1$ and that of class $\omega=2$ respectively. Figure 6-3(c) shows a superposition of Figures 6-3(a) and (b). Figure 6-3(d) shows the PoCM computed using equation (3.33).

Figures 6-3(a) and (b) show that the joint posterior probability of a particular class are highest at the region where there are more training samples of the same class. Figure 6-3(d) shows that the samples from patients with diabetes ($\omega=2$) and normal patients ($\omega=1$) are not well separated, and a complex decision boundary is generated, which is similar to Figure 6-1(d). The results show that there are a large number of samples from both classes at the complex decision boundary. Hence, the width of the uncertainty region at the complex decision boundary is significantly large. This result reinforces the studies presented in section 4.5 and shows that the proposed PoCM is suitable for estimating the classification confidence of modeling the Pima Indian diabetes dataset so that classifications with low confidence can be rejected.

In the next part of the experiment, the PrCM of NBPW and PoCM of RNFS are used to reject the classifications with low confidence to improve the classification accuracies. A sample $\mathbf{X}$ is rejected if the confidence measure of the sample $CM(\mathbf{X})$ is less than a defined confidence threshold. Figure 6-4 shows a plot of e and r against the PrCM $CM(\mathbf{X})$ obtained from NBPW on the test data.

Figure 6-4: Plot of the error rate e and rejection rate r against the PrCM of NBPW on the Pima Indians diabetes dataset

The error rate e is defined as

$$e = \frac{n_M}{n_A} \quad (6.1)$$

where n_M is the number of misclassified test samples and n_A is the number of accepted test samples. The reject rate r is defined as

$$r = \frac{n_R}{n} \quad (6.2)$$

where n_R is the number of test samples rejected and $n = n_A + n_R$ is the total number of test samples.

Figure 6-4 shows that if the threshold of $CM(X)$ of rejecting the samples is increased, the reject rate r increases monotonically, but the error rate e does not decrease monotonically. This result is consistent with Chow's theory that the tradeoff between errors and rejects are seldom one for one; since whenever the reject option is exercised, some would-be correct classifications are also rejected (Chow, 1970). Therefore, an optimal rejection confidence threshold that will result in the best error-reject tradeoff is desired. By Chow's definition, an optimal recognition rule is one that minimizes the reject rate for a given error rate (Chow, 1970). However, finding an optimal rejection confidence threshold of a classification task is a subjective matter,

which depends on whether the misclassification of a sample is more costly than the rejection of the sample. This subjective threshold can be expressed objectively as a function

$$J = \alpha e^2 + r^2, \quad (6.3)$$

where α represents a ratio on the cost of misclassification versus rejection, e is the error rate and r is the rejection rate. Using this objective function, the optimal rejection confidence threshold can be obtained from the point where the objective function is minimized.

Figures 6-5(a) and (b) show plots of the objective functions $J=2e^2+r^2$ and $J=4e^2+r^2$ against the PrCM of NBPW respectively. The results show that the former yielded an optimal rejection PrCM of 19.4% and the latter, which represents a heavier cost on misclassifications, yielded 52.9%.

Figure 6-5: Plot of the error rate e and rejection rate r against the PrCM $CM(X)$ of NBPW on the test data of the Pima Indians diabetes dataset using a objective function of (a) $J=2e^2+r^2$; and (b) $J=4e^2+r^2$

Table 6-4 shows the experimental results on the confusion matrix of using NBPW with optimal rejection using PrCM on the test data of the Pima Indians diabetes dataset. As a benchmark, Li and Sethi's (2006b) experimental results yielded a test accuracy of $(53+12)/(122+70-67-57)=95.59\%$ that rejected classifications based on the output scores of SVM.

Table 6-4: Confusion matrix on the test data of the Pima Indians diabetes dataset using NBPW with optimal rejection using PrCM

Observation	Confidence Threshold of 19.4%				Confidence Threshold of 52.9%			
	Prediction		Reject	Total	Prediction		Reject	Total
	Negative	Positive			Negative	Positive		
Negative	100	11	11	122	79	4	39	122
Positive	15	43	12	70	4	23	43	70

The results in Table 6-4 show that the number of misclassifications is significantly reduced, and the test accuracy improved from 80.73% to $(100+43)/(122+70-11-12)=84.62\%$. In addition, when the objective function that represents a heavier cost on misclassification is used, the test accuracy improved further to 92.73%. The results show that NBPW that employed PrCM to reject classifications with a low confidence threshold of 52.9% yielded slightly poorer test accuracies than Li and Sethi's (2006b) results that reject classifications based on the output scores of SVM. However, the former rejected $39+43=82$ samples compared against $67+57=124$ in the latter. Thus, the latter yielded slighter better test accuracy at a consequence of higher rejections.

Figures 6-6(a) and (b) show plots of the objective functions $J=2e^2+r^2$ and $J=4e^2+r^2$ respectively against the PoCM of RNFS. The results show that the former yielded an optimal rejection PoCM of 6.42% and the latter, which represents a heavier cost on misclassifications, yielded 12.8%.

Figure 6-6: Plot of the error rate e and rejection rate r against the PoCM $CM(X)$ of RNFS on the test data of the Pima Indians diabetes dataset using a objective function of (a) $J=2e^2+r^2$; and (b) $J=4e^2+r^2$

Table 6-5 shows the experimental results on the confusion matrix of using RNFS with optimal rejection using PoCM on the test data of the Pima Indians diabetes dataset.

Table 6-5: Confusion matrix on the test set of the Pima Indians diabetes dataset using RNFS with optimal rejection using PoCM

Observation	Confidence Threshold of 6.42%				Confidence Threshold of 12.8%			
	Prediction		Reject	Total	Prediction		Reject	Total
	Negative	Positive			Negative	Positive		
Negative	92	17	13	122	76	9	37	122
Positive	17	37	16	70	14	31	25	70

The results in Table 6-5 show that the number of misclassifications has been significantly reduced and accuracies improved from 72.92% to $(92+37)/(122+70-13-16)=79.14\%$. In addition, when the objective function that represents a heavier cost on misclassification is used, the classification accuracy improved further to 82.31%. The results show that RNFS that employed PoCM to reject classifications with a low confidence threshold of 12.8% yielded poorer test accuracies than NBPW that employed PrCM and Li and Sethi's (2006b) results. However, the RNFS that employed PoCM rejected $37+25=62$ samples compared against 82 by NBPW that employed PrCM and 124 by Li and Sethi's (2006b).

Comparing Tables 6-4 and 6-5 with Tables 6-1 and 6-3, the results show that the number of misclassifications is significantly reduced and accuracies improved by rejecting classifications with low confidence. In addition, when the objective function that represents a heavier cost on misclassification is used, the classification accuracy improved further. Further improvements to the classification accuracies can be achieved using an objective function that represents an increased cost on misclassifications, but the consequence is a further increase in the number of rejections. Although NBPW that employed PrCM yielded better accuracy than RNFS that employed PoCM, the latter yielded a model with higher interpretability.

However, the result from one set of training and test data is not sufficiently conclusive. Therefore, in the last part of this set of the experiments, 3-fold cross-validations with selection bias are performed on the Pima Indian diabetes dataset. Figures 6-7(a) and (b) show the plots of the objective functions $J=2e^2+r^2$ and $J=4e^2+r^2$ against the PrCM of NBPW respectively on the 3-fold cross-validation sets of the Pima Indian diabetes dataset. The results show that the former yielded an optimal rejection PrCM of 26.2% and the latter yielded 61.8%. Table 6-6 shows the experimental results on the confusion matrix of using NBPW with optimal rejection using PrCM on the 3-fold cross-validation sets of the Pima Indians diabetes dataset.

Figure 6-7: Plot of the error rate e and rejection rate r against the PrCM $CM(x)$ of NBPW on the 3-fold cross-validation sets of the Pima Indians diabetes dataset using a objective function of (a) $J=2e^2+r^2$; and (b) $J=4e^2+r^2$

Table 6-6: Confusion matrix on the cross-validation sets of the Pima Indians diabetes dataset using NBPW with optimal rejection using PrCM

PrCM Threshold	Observation	CV1			CV2			CV3			Total
		Prediction		Reject	Prediction		Reject	Prediction		Reject	
26.2%	Negative	128	8	31	133	17	17	125	11	30	500
	Positive	18	45	26	26	44	19	21	44	25	268
61.8%	Negative	96	3	68	96	8	63	97	3	66	500
	Positive	5	29	55	8	27	54	7	16	67	268

Figures 6-8(a) and (b) show the plots of the objective functions $J=2e^2+r^2$ and $J=4e^2+r^2$ against the PoCM of RNFS respectively on the 3-fold cross-validation sets of the Pima Indian diabetes dataset. The results show that the former yielded an optimal rejection PoCM of 7.54% and the latter yielded 21.9%. Table 6-7 shows the experimental results on the confusion matrix of using NBPW with optimal rejection

using PrCM on the 3-fold cross-validation sets of the Pima Indians diabetes dataset.

Figure 6-8: Plot of the error rate e and rejection rate r against the PoCM $CM(X)$ of RNFS on the 3-fold cross-validation sets of the Pima Indians diabetes dataset using a objective function of (a) $J=2e^2+r^2$; and (b) $J=4e^2+r^2$

Table 6-7: Confusion matrix on the cross-validation sets of the Pima Indians diabetes dataset using RNFS with optimal rejection using PoCM

		CV1			CV2			CV3			
PoCM		Prediction			Prediction			Prediction			
Threshold	Observation	Negative	Positive	Reject	Negative	Positive	Reject	Negative	Positive	Reject	Total
	Negative	123	17	27	116	26	25	132	18	16	500
7.54%	Positive	26	50	13	21	55	13	31	41	18	268
	Negative	87	10	70	84	17	66	98	7	61	500
21.9%	Positive	11	38	40	10	39	40	12	26	52	268

Table 6-8: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the Pima Indians diabetes dataset

Classifier	Without rejection	With rejection $J=2e^2+r^2$		With rejection $J=4e^2+r^2$	
	Accuracy (%)	CM Threshold	Accuracy (%)	CM Threshold	Accuracy (%)
NBPW	77.86	26.2	83.82	7.54	91.44
RNFS	75.65	61.8	78.82	21.9	84.78

Table 6-8 shows the consolidated results on the accuracies of using NBPW and RNFS without rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the Pima Indian diabetes dataset. The results show that the accuracies improved significantly when the classifications with low confidences are rejected using the objective function $J=2e^2+r^2$. When the objective function $J=4e^2+r^2$ that represents a heavier cost on misclassification is used, the accuracies improved further. Thus, the results show that the use of PrCM and PoCM to reject classifications with low confidence improved the accuracies for NBPW and RNFS respectively. Although NBPW yielded a superior accuracy than RNFS on the Pima Indian diabetes dataset, the latter yielded a model with higher interpretability.

6.3 Classifying high dimensional biomedical data

This section investigates the effectiveness of using RNFS and the proposed PoCM to reject classifications with low confidence on high dimensional biomedical data. The effectiveness is investigated in terms of the accuracy in the classification of the test data. The performance of the RNFS employing the proposed MIRS feature selection using wrapper approach without rejecting classification with low confidence is compared against a selection of classifiers; namely NBPW, FLD, CART, k-NN, SVM and DENFIS. The selection of classifiers employed a selection of MI-based feature selection algorithms and a rough set-based feature selection algorithm; namely, the New Fuzzy-Rough Feature Selection (FRFS2) algorithm (Jensen & Shen, 2007b). The performance of the RNFS with rejection of classifications with low confidence using the proposed PoCM is compared against the NBPW using PrCM.

The effectiveness of RNFS in performing feature selection and classification on high dimensional dataset is first evaluated using 3-fold and 10-fold cross-validations without selection bias and without rejections. This is similar to section 4.4.1 in which the experiment is repeated in three and ten stages respectively for 3-fold and 10-fold cross-validations using different permutations of the training and test data sets and different classifiers using different feature selection algorithms. Next, the effectiveness of using the proposed PoCM of RNFS is compared against PrCM of NBPW using 3-fold cross-validations with selection bias and with rejections of classifications with low confidence. A selection bias is eliminated if feature selection is performed in the training process of the classifiers at each stage of the cross-validations, which tend to yield results with relatively lower accuracies than cross-validations with selection bias (Ambroise & McLachlan, 2002). This part focuses on evaluating the use of the proposed confidence measures to reject classifications with

low confidence to improve the classification accuracies and not on the cross-validation accuracies of the feature selection method. Hence, the simpler method of performing cross-validations with selection bias suffices. The proposed PoCM of RNFS is only compared against the PrCM of NBPW since the other classifiers are not capable of rejecting classifications with low confidence.

Three sets of experiments are performed to investigate the effectiveness of RNFS in modeling high dimensional biomedical data; namely the Bhattacharjee lung cancer dataset (Bhattacharjee et al., 2001), the Schummer ovarian cancer dataset (Schummer et al., 1999), and the Golub Leukemia dataset (Golub et al., 1999). These experiments are performed using Mathworks Matlab R2007a on a Dell Optiplex GX745 PC equipped with Intel Core2 Duo 2.4 Ghz and 2 Gb RAM.

6.3.1 Experiment: Bhattacharjee lung cancer dataset

In this section, the Bhattacharjee lung cancer mRNA microarray gene expression dataset B (Bhattacharjee et al., 2001) is used, which consists of a total of 156 tissue samples with 675 features where 139 samples are from patients with adenocarcinomas lung cancer and 17 samples are normal patients. Each feature represents one of the most variably expressed gene transcript sequence from a patient's tissue sample (Bhattacharjee et al., 2001).

6.3.1.1 Feature selection and classification

Table 6-9 presents the consolidated experimental results of the training and test accuracies performed using 3-fold and 10-fold cross-validations without selection bias and without rejection for the first part of the experiments. The average accuracies of using the selection of classifiers and feature selection algorithms are presented. Figure 6-9 shows a plot of the test classification accuracies against the selection of classifiers and feature selection algorithms.

Table 6-9: Consolidated experimental results of the classification accuracies of using a selection of classifiers and a selection of feature selection algorithms performed using 3-fold and 10-fold cross-validations without selection bias and without rejection on the Bhattacharjee lung cancer dataset

Validation Method				3-fold cross-validation		10-fold cross-validation			
Feature Selection	Features	Exec (s)	Classifier	Training Acc(%)	Test Acc(%)	Features	Training Acc(%)	Test Acc(%)	
Filter Approach	MIBIFPW	k = 3	NBPW	94.2 ± 5.4	92.9 ± 2.9	k = 3.00	97.1 ± 2.8	97.5 ± 4.4	
			FLD	79.5 ± 18.8	75.6 ± 18.7		94.2 ± 11.5	94.3 ± 13.6	
			CART	95.8 ± 6.4	91.0 ± 1.1		97.8 ± 3.0	96.8 ± 5.3	
			3-NN	88.8 ± 14.5	84.0 ± 16.4		95.7 ± 8.8	95.0 ± 11.7	
			SVM	93.3 ± 5.4	92.9 ± 2.9		96.0 ± 2.5	97.5 ± 4.4	
			RNFS	66.3 ± 48.6	91.7 ± 1.1		88.8 ± 27.4	96.8 ± 4.5	
			DENFIS	88.5 ± 1.9	89.7 ± 1.1		96.6 ± 2.7	97.5 ± 4.4	
		k = 6	NBPW	95.5 ± 6.1	94.9 ± 4.0	k = 6	97.8 ± 3.0	96.8 ± 5.5	
			FLD	87.2 ± 20.5	81.4 ± 27.2		94.7 ± 11.7	93.7 ± 13.9	
			CART	97.1 ± 5.0	92.9 ± 4.8		98.1 ± 2.1	94.3 ± 8.6	
			3-NN	95.5 ± 6.1	93.6 ± 6.2		97.9 ± 2.6	95.5 ± 8.5	
			SVM	94.9 ± 5.6	94.2 ± 3.3		96.6 ± 2.7	96.8 ± 5.5	
			RNFS	95.8 ± 6.4	94.2 ± 3.8		97.7 ± 3.0	96.1 ± 5.5	
			DENFIS	95.5 ± 6.2	93.6 ± 2.9		93.3 ± 5.0	91.8 ± 8.9	
	MIPW	k = 9	NBPW	95.5 ± 6.1	94.2 ± 3.3	k = 9	97.4 ± 2.9	96.1 ± 5.5	
			FLD	88.5 ± 17.5	83.3 ± 20.6		94.9 ± 10.3	93.7 ± 13.5	
			CART	98.7 ± 2.2	94.2 ± 1.9		98.8 ± 1.2	94.9 ± 8.8	
			3-NN	97.1 ± 3.3	96.2 ± 1.9		98.2 ± 1.7	95.5 ± 8.5	
			SVM	95.5 ± 6.1	94.9 ± 4.0		97.0 ± 2.8	96.8 ± 5.5	
			RNFS	96.2 ± 6.7	94.9 ± 4.8		98.2 ± 3.2	96.1 ± 5.5	
			DENFIS	95.8 ± 6.4	91.7 ± 6.2		97.6 ± 3.3	97.5 ± 4.4	
		k = 12	NBPW	96.2 ± 3.3	90.4 ± 6.9	k = 12	97.5 ± 2.2	94.8 ± 7.3	
			FLD	90.7 ± 13.6	85.3 ± 17.2		94.5 ± 9.7	91.0 ± 15.1	
			CART	100.0 ± 0.0	91.7 ± 4.8		99.5 ± 0.9	96.2 ± 6.7	
			3-NN	98.4 ± 1.1	94.9 ± 4.0		98.4 ± 1.2	96.8 ± 5.5	
			SVM	99.0 ± 1.0	96.2 ± 1.9		97.7 ± 3.0	96.8 ± 5.5	
			RNFS	98.7 ± 2.2	93.6 ± 6.8		98.8 ± 2.4	96.1 ± 5.5	
			DENFIS	98.4 ± 2.8	91.0 ± 8.9		98.6 ± 3.3	96.2 ± 5.4	
	MIPW	2.0 ± 0.0	NBPW	99.4 ± 1.1	97.4 ± 1.1	2.0 ± 0.0	98.1 ± 1.2	96.8 ± 5.5	
			FLD	98.4 ± 0.6	96.8 ± 2.2		97.8 ± 0.9	96.8 ± 5.5	
			CART	100.0 ± 0.0	96.2 ± 3.8		99.2 ± 0.5	94.8 ± 5.1	
			3-NN	98.7 ± 0.6	96.2 ± 1.9		98.5 ± 0.5	96.8 ± 5.5	
			SVM	98.4 ± 0.6	95.5 ± 2.9		97.3 ± 1.2	96.8 ± 5.5	
			RNFS	10.9 ± 0.6	94.9 ± 4.4		28.2 ± 36.5	94.2 ± 6.3	
			DENFIS	100.0 ± 0.0	96.2 ± 1.9		99.6 ± 0.4	95.5 ± 6.1	
		FRFS2	2.3 ± 0.6	NBPW	99.4 ± 1.1	97.4 ± 1.1	2.8 ± 0.4	98.7 ± 0.7	97.4 ± 4.6
				FLD	98.1 ± 0.0	97.4 ± 1.1		97.8 ± 0.9	95.5 ± 4.3
				CART	100.0 ± 0.0	95.5 ± 4.0		99.1 ± 0.8	96.8 ± 4.5
				3-NN	98.4 ± 0.6	96.8 ± 1.1		98.3 ± 0.9	96.8 ± 4.5
				SVM	98.7 ± 0.6	96.8 ± 1.1		98.0 ± 0.9	96.2 ± 4.4
				RNFS	40.4 ± 51.6	93.6 ± 5.9		28.4 ± 37.0	95.5 ± 6.0
				DENFIS	100.0 ± 0.0	96.8 ± 1.1		99.4 ± 0.6	94.3 ± 4.6
Wrapper Approach	MINBPW	2.0 ± 0.0	6.2 ± 0.1	NBPW	95.2 ± 5.0	95.5 ± 6.2	2.0 ± 0.0	98.5 ± 0.4	98.1 ± 3.1
MIRSR	6.3 ± 2.5	7.9 ± 0.2	RNFS	100.0 ± 0.0	94.2 ± 0.0	9.5 ± 2.7	100.0 ± 0.0	98.1 ± 3.1	

The results show that NBPW that employed MIPW or FRFS2 and FLD that employed FRFS2 using the filter approaches yielded a superior test accuracy of 97.4±1.1 on 3-fold cross-validations. The NBPW that employed MINBPW and the RNFS that employed the proposed MIRSR, both using the wrapper approach, yielded superior test accuracy of 98.1±3.1 on 10-fold cross-validations. The results show that the experimental setup in the choice of k-fold cross-validation affects the on the test

accuracy. Nevertheless, the results show that both the NBPW that employed MINBPW and RNFS that employed the proposed MIRSRS yielded relatively good test accuracies of approximately 95% without rejecting classifications with low confidence compared against the other classifiers and feature selection algorithms.

(a)

(b)

Figure 6-9: Consolidated results on the classification accuracies performed using (a) 3-fold; and (b) 10-fold cross-validations without selection bias and without rejection on the Bhattacharjee lung cancer dataset

6.3.1.2 Estimation of classification confidence and rejection

Table 6-10 shows the experimental results on the confusion matrix of using NBPW that employed MINBPW on 3-fold cross-validation sets for the second part of the experiments. The MIBNPW selected 2 of the 675 features in the dataset, namely the 388th feature on gene “AF070648” and the 515th feature on gene “AL096719”.

Table 6-10: Confusion matrix on the 3-fold validation sets of the Bhattacharjee lung cancer dataset using NBPW without rejection

	CV1		CV2		CV3		
	Prediction		Prediction		Prediction		
Observation	Negative	Positive	Negative	Positive	Negative	Positive	Total
Negative	4	2	6	0	5	0	17
Positive	0	46	1	45	0	47	139

Table 6-10 shows that among the 139 patients with adenocarcinomas lung cancer in the three cross-validation sets, only 1 is misclassified. Although the result is apparently acceptable for medical diagnosis, the rejection of classifications with low confidence is investigated for improvements in accuracy.

Figure 6-10 shows the surface plot of the joint posterior probability and PrCM of NBPW against the FLD transformed features on one of the three cross-validation sets.

Figure 6-10(d) shows that the samples from patients with adenocarcinomas lung cancer ($\omega=2$) and normal patients ($\omega=1$) are well separated with a complex decision boundary. The results show that there are a small number of samples from both classes at the complex decision boundary. Hence, the width of the uncertainty region at the complex decision boundary is significantly small. This result shows that the proposed PrCM is suitable for estimating the classification confidence of modeling the Bhattacharjee lung cancer dataset.

Figure 6-10: Plot of the joint probability (a) $p(X|\omega_1)$ for $\omega=1$, (b) $p(X|\omega_2)$ $\omega=2$, (c) $p(X|\omega_i)$ for both $\omega=1$ and 2 and (d) plot of the PrCM $CM(X)$ of NBPW against FLD transformed features t_1 and t_2 of one of the three cross-validation sets from the Bhattacharjee lung cancer dataset

Figure 6-11: Plot of the membership functions generated using the proposed SPSEC employed in RNFS that characterize the possibility distributions (a) $\pi_{\omega|X}(X_{388}, \omega)$; (b) $\pi_{\omega|X}(X_{393}, \omega)$; and (c) $\pi_{\omega|X}(X_{270}, \omega)$ of one of the three cross-validation sets from the Bhattacharjee lung cancer dataset

Next, the experiment is repeated using RNFS. Figure 6-11 shows the membership functions generated using the proposed SPSEC employed in RNFS, and the proposed MIRS employed in RNFS selected 11 out of the 675 features in the dataset. These

11 features are further reduced to 3 features by the proposed RSPOP employed in RNFS; namely, the 388th feature on gene “AF070648”, the 393rd feature on gene “AL031058”, and the 270th feature on gene “AL049963”. Table 6-11 shows the if-then fuzzy rules identified using the proposed RSPOP employed in RNFS.

Table 6-11: If-then fuzzy rules identified using the proposed RSPOP employed in RNFS on one of the cross-validation sets from the Bhattacharjee lung cancer dataset

R_1	IF AF070648 is small AND AL031058 is large
R_2	IF AL031058 is medium AND AL049963 is small
R_4	IF AF070648 is small AND AL031058 is medium AND AL049963 is large
R_5	IF AF070648 is large AND AL031058 is large AND AL049963 is large THEN patient is cancerous
R_3	IF AF070648 is large AND AL031058 is medium AND AL049963 is large THEN patient is normal

The results show that the membership functions that characterize the possibility distributions in Figure 6-11 resemble human interpretable linguistic terms and the if-then fuzzy classification rules in Table 6-11 are humanly interpretable. Therefore, the results show that RNFS yielded a model with higher interpretability than NBPW. Table 6-12 shows the experimental results on the confusion matrix of using RNFS that employed MIRSR on the 3-fold cross-validation sets.

Table 6-12: Confusion matrix on the 3-fold validation sets of the Bhattacharjee lung cancer dataset using RNFS without rejection

Observation	CV1		CV2		CV3		Total
	Prediction		Prediction		Prediction		
	Negative	Positive	Negative	Positive	Negative	Positive	
Negative	5	1	6	0	5	0	17
Positive	1	45	1	45	3	44	139

Table 6-12 shows that among the 139 patients with adenocarcinomas lung cancer in the validation sets, 5 are misclassified. Therefore, the results yielded slightly poorer test accuracy than NBPW that employed MINBPW on the 3-fold cross-validation sets.

Figure 6-12 shows the surface plot of the joint possibility and PoCM of RNFS against the FLD transformed features on one of the three cross-validation sets.

Figure 6-12: Plot of the joint possibility distribution (a) $\pi_{\omega_1, \lambda}(X, \omega_1)$ for $\omega=1$, (b) $\pi_{\omega_2, \lambda}(X, \omega_2)$ for $\omega=2$, (c) $\pi_{\omega, \lambda}(X, \omega)$ for both $\omega=1$ and 2 and (d) plot of the PoCM $CM(X)$ of RNFS against transformed features t_1 and t_2 of one of the cross-validation sets from the Bhattacharjee lung cancer dataset

Figure 6-12(d) shows that the patients with adenocarcinomas lung cancer ($\omega=2$) and normal patients ($\omega=1$) are not very well separated, and a complex decision boundary is generated. The results show that there are a small number of samples from both classes at the complex decision boundary. Hence, the width of the uncertainty region at the complex decision boundary is significantly small. This result shows that the proposed PoCM is suitable for estimating the classification confidence of modeling the Bhattacharjee lung cancer dataset.

In the next part of the experiment, the PrCM of NBPW and PoCM of RNFS are used to reject the classifications with low confidence to improve the classification accuracies. Figures 6-13(a) and (b) show the plot of the objective function $J=4e^2+r^2$ against the PrCM of NBPW and PoCM of RNFS respectively. The results show that the former yielded an optimal rejection PrCM threshold of 94.3% and the latter yielded an optimal rejection PoCM threshold of 49.4%. Table 6-13 shows the experimental results on the confusion matrix of using NBPW and RNFS with optimal rejection using PrCM and PoCM on the 3-fold cross-validation sets.

Figure 6-13: Plot of the error rate e and rejection rate r against (a) PrCM $CM(X)$ of NBPW and (b) PoCM $CM(X)$ of RNFS on the 3-fold cross-validation sets of the Bhattacharjee lung cancer dataset using an objective function of $J=4e^2+r^2$

Table 6-13: Confusion matrix on the cross-validation sets of the Bhattacharjee lung cancer dataset using NBPW and RNFS with optimal rejection using PrCM and PoCM respectively

		CV1			CV2			CV3			
		Prediction			Prediction			Prediction			
	Observation	Negative	Positive	Reject	Negative	Positive	Reject	Negative	Positive	Reject	Total
NBPW with PrCM	Negative	3	0	3	5	0	1	4	0	1	17
Threshold 94.3%	Positive	0	45	1	0	44	2	0	47	0	139
RNFS with PoCM	Negative	5	0	1	4	0	2	3	0	2	17
Threshold 49.4%	Positive	1	45	0	0	44	2	0	44	3	139

Table 6-14: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the Bhattacharjee lung cancer dataset

Classifier	Without rejection	With rejection	$J=4e^2+r^2$
	Accuracy (%)	CM Threshold (%)	Accuracy (%)
NBPW	98.08	94.3	100
RNFS	96.15	49.4	99.35

Table 6-14 shows the consolidated results on the accuracy of using NBPW and RNFS without rejections versus with optimal rejection using PrCM and PoCM respectively on the 3-fold cross-validation sets of the Bhattacharjee lung cancer dataset. Although the classification accuracies are significantly high without rejections, the results show that the classification accuracies improved further when classification with low confidence are rejected using the objective function $J=4e^2+r2$. Thus, the results show that the use of PrCM and PoCM to perform optimal rejection improved the classification accuracies of NBPW and RNFS respectively. Although NBPW yielded a superior accuracy than RNFS on the Bhattacharjee lung cancer dataset, most importantly, the latter yielded a model with higher interpretability.

6.3.2 Experiment: Schummer ovarian cancer dataset

In this section, the Schummer ovarian cancer DNA microarray gene expression dataset (Schummer et al., 1999) is used, which consists of a total of 54 tissue samples with 1,536 features where 30 samples are from patients with ovarian tumor and 24 samples are from normal patients. Each feature represents a position of the 1,536 dots of a phosphor image from a membrane that was hybridized with the cDNA strand from a patient's tissue sample (Schummer et al., 1999).

6.3.2.1 Feature selection and classification

Table 6-15 presents the consolidated experimental results of the training and test accuracies performed using 3-fold and 10-fold cross-validations without selection bias for the first part of the experiments. The average accuracies of using the selection of classifiers and feature selection algorithms are presented. Figure 6-14 shows a plot of the test classification accuracies against the selection of classifiers and feature selection algorithms.

Table 6-15: Consolidated experimental results of the classification accuracies of using a selection of feature selection algorithms and a selection of classifiers performed using 3-fold and 10-fold cross validations without selection bias and without rejection on the Schummer ovarian cancer dataset

Validation Method				3-fold cross-validation		10-fold cross-validation			
Feature Selection	Features	Exec (s)	Classifier	Training Acc(%)	Test Acc(%)	Features	Training Acc(%)	Test Acc(%)	
Filter Approach	MIBIFPW	k = 3	4.4 ± 0.0	NBPW	70.4 ± 20.8	68.5 ± 22.5	k = 3.00	61.3 ± 11.5	58.0 ± 9.2
				FLD	67.6 ± 23.3	63.0 ± 27.4		58.4 ± 11.8	45.7 ± 12.9
				CART	90.7 ± 5.8	59.3 ± 25.7		85.4 ± 4.9	55.7 ± 16.6
				3-NN	85.2 ± 11.6	81.5 ± 6.4		85.2 ± 3.5	68.7 ± 23.4
				SVM	71.3 ± 20.1	68.5 ± 17.9		62.9 ± 10.9	59.7 ± 9.1
				RNFS	56.5 ± 10.5	70.4 ± 16.0		66.2 ± 9.1	58.0 ± 12.1
				DENFIS	72.2 ± 19.4	68.5 ± 19.5		67.1 ± 9.0	59.7 ± 13.1
		k = 6	4.4 ± 0.0	NBPW	84.3 ± 20.5	79.6 ± 22.5	k = 6	86.1 ± 12.7	79.3 ± 14.1
				FLD	81.5 ± 22.6	79.6 ± 26.3		84.2 ± 12.4	77.3 ± 18.0
				CART	97.2 ± 2.8	77.8 ± 14.7		95.9 ± 2.7	81.0 ± 16.4
				3-NN	92.6 ± 10.5	75.9 ± 22.5		93.0 ± 3.6	82.7 ± 19.9
				SVM	87.0 ± 20.1	81.5 ± 22.5		87.7 ± 12.6	81.0 ± 18.9
				RNFS	89.8 ± 15.3	87.0 ± 17.9		90.6 ± 12.6	87.3 ± 11.8
				DENFIS	89.8 ± 15.3	83.3 ± 5.6		89.4 ± 12.3	85.0 ± 12.4
	MIPW	k = 9	4.4 ± 0.0	NBPW	92.6 ± 1.6	88.9 ± 9.6	k = 9	89.9 ± 2.4	89.3 ± 12.2
				FLD	91.7 ± 0.0	88.9 ± 5.6		89.9 ± 2.4	89.3 ± 12.2
				CART	98.1 ± 3.2	87.0 ± 3.2		96.5 ± 2.6	93.0 ± 9.1
				3-NN	97.2 ± 4.8	88.9 ± 0.0		97.3 ± 2.0	96.7 ± 7.0
				SVM	97.2 ± 4.8	94.4 ± 5.6		97.1 ± 2.6	92.7 ± 9.5
				RNFS	98.1 ± 3.2	96.3 ± 3.2		98.1 ± 1.5	92.7 ± 9.5
				DENFIS	95.4 ± 5.8	81.5 ± 17.9		96.9 ± 2.5	91.3 ± 12.1
		k = 12	4.4 ± 0.0	NBPW	92.6 ± 1.6	90.7 ± 3.2	k = 12	90.9 ± 2.0	87.3 ± 18.8
				FLD	91.7 ± 0.0	90.7 ± 3.2		90.5 ± 2.2	87.3 ± 18.8
				CART	98.1 ± 3.2	87.0 ± 3.2		97.3 ± 1.9	92.7 ± 9.5
				3-NN	95.4 ± 5.8	88.9 ± 5.6		97.7 ± 0.6	95.0 ± 11.2
				SVM	97.2 ± 4.8	94.4 ± 5.6		97.5 ± 1.9	91.0 ± 12.4
				RNFS	97.2 ± 4.8	94.4 ± 5.6		98.8 ± 1.1	92.3 ± 9.9
				DENFIS	100.0 ± 0.0	79.6 ± 8.5		97.1 ± 1.7	82.3 ± 20.0
	FRFS2	2.7 ± 0.6	23.9 ± 3.3	NBPW	98.1 ± 3.2	92.6 ± 3.2	3.1 ± 0.3	96.5 ± 2.2	91.0 ± 12.4
				FLD	97.2 ± 4.8	90.7 ± 3.2		92.8 ± 3.8	83.3 ± 16.6
				CART	98.1 ± 1.6	92.6 ± 3.2		96.1 ± 2.6	91.3 ± 12.1
				3-NN	97.2 ± 2.8	92.6 ± 6.4		95.9 ± 2.2	87.3 ± 15.1
				SVM	98.1 ± 3.2	94.4 ± 5.6		95.1 ± 2.4	87.7 ± 14.1
				RNFS	96.3 ± 3.2	88.9 ± 5.6		95.3 ± 2.9	91.0 ± 15.6
				DENFIS	100.0 ± 0.0	94.4 ± 5.6		98.8 ± 1.5	91.0 ± 12.4
		2.7 ± 0.6	0.7 ± 0.1	NBPW	98.1 ± 3.2	88.9 ± 5.6	3.0 ± 0.0	95.5 ± 2.9	91.0 ± 13.4
				FLD	96.3 ± 4.2	87.0 ± 6.4		93.0 ± 3.7	91.0 ± 13.4
				CART	99.1 ± 1.6	90.7 ± 3.2		97.3 ± 1.9	91.3 ± 12.1
				3-NN	98.1 ± 1.6	85.2 ± 8.5		96.5 ± 2.7	87.7 ± 16.9
				SVM	96.3 ± 6.4	88.9 ± 5.6		95.3 ± 2.8	91.3 ± 12.1
				RNFS	78.7 ± 29.7	83.3 ± 5.6		87.6 ± 15.5	93.3 ± 11.7
				DENFIS	99.1 ± 1.6	85.2 ± 8.5		98.8 ± 1.1	89.3 ± 15.4
Wrapper Approach	MINBPW	2.3 ± 0.6	6.5 ± 0.5	NBPW	94.4 ± 2.8	88.9 ± 5.6	2.7 ± 0.5	94.5 ± 2.9	83.0 ± 13.7
MIRSR	7.7 ± 2.1	7.9 ± 0.1	RNFS	100.0 ± 0.0	92.6 ± 3.2	6.8 ± 1.9	94.5 ± 17.4	92.3 ± 9.9

(a)

(b)

Figure 6-14: Consolidated results on the classification accuracies performed using (a) 3-fold; and (b) 10-fold cross-validations without selection bias and without rejection on the Schummer ovarian cancer dataset

The results show that RNFS that employed MIBIFPW using the filter approach yielded a superior test accuracy of $96.3\pm3.2\%$ on 3-fold cross-validations. The 3-NN that employed MIBIFPW using the filter approach yielded a superior test accuracy of 96.7 ± 7.0 on 10-fold cross-validations. Nevertheless, the RNFS that employed the MIRSR using the wrapper approach yielded a relatively good test accuracies of

92.6±3.2% and 92.3±9.9% without rejecting classifications with low confidence. However, the NBPW that employed the MINBPW using the wrapper approach yielded a relatively poorer test accuracies of 88.9±5.6% and 83.0±3.2% without rejecting classifications with low confidence. Comparing the results in Table 6-15 with Table 6-9, superior test accuracies are obtained from different classifiers employing different feature selection algorithm on different experiment setup. This shows that there is no general superiority of any models in the modeling of classification data. This is known as the *No Free Lunch Theorem* in pattern recognition literature (Duda et al., 2001), which is interpreted as when a model outperforms others on a particular dataset, it is a consequence of its fit to the particular dataset and not the general superiority of the model.

6.3.2.2 Estimation of classification confidence and rejection

Table 6-16 shows the experimental results on the confusion matrix of using the NBPW that employed MINBPW on 3-fold cross-validations sets for the second part of the experiments. The MINBPW selected 3 of the 1,536 features in the dataset, namely the 9th, 272nd and 930th dot.

Table 6-16: Confusion matrix on the 3-fold validation sets of the Schummer ovarian cancer dataset using NBPW without rejection

Observation	CV1		CV2		CV3		
	Prediction		Prediction		Prediction		
	Negative	Positive	Negative	Positive	Negative	Positive	Total
Negative	8	0	8	0	8	0	24
Positive	1	9	1	9	2	8	30

Table 6-16 shows that among the 30 patients with ovarian cancer in the validation sets, 4 are misclassified.

Figure 6-15 shows the surface plot of the joint posterior probability and the PrCM against the FLD transformed features on one of the three cross-validation sets.

Figure 6-15: Plot of the joint posterior probability distribution (a) $p(X|\omega_1)$ for $\omega=1$, (b) $p(X|\omega_2)$ for $\omega=2$, (c) $p(X|\omega)$ for both $\omega=1$ and 2 and (d) plot of the PrCM $CM(X)$ against FLD transformed features t_1 and t_2 of one of the three cross-validation sets from the Schummer ovarian cancer dataset

Figure 6-15(d) shows that the samples from patients with ovarian cancer ($\omega=2$) and normal patients ($\omega=1$) are not very well separated with a complex decision boundary. The results show that there are a small number of samples from both classes at the complex decision boundary. Hence, the width of the uncertainty region at the complex decision boundary is significantly small. This result shows that the proposed PrCM is suitable for estimating the classification confidence of modeling the Schummer ovarian cancer dataset.

Next, the experiment is repeated using RNFS. Figure 6-16 shows the membership functions generated using the proposed SPSEC employed in RNFS, and the proposed

MIRSR employed in RNFS selected 6 of the 1,536 features in the dataset. These 6 features are further reduced to 4 features by the proposed RSPOP employed in RNFS; namely, the 1483rd dot, the 9th dot, the 93rd dot, and the 1526th dot. Table 6-17 shows the if-then fuzzy rules identified using the proposed RSPOP employed in RNFS.

Figure 6-16: Plot of the membership functions generated using the proposed SPSEC employed in RNFS that characterize the possibility distributions (a) $\pi_{\omega|X}(X_{1483}, \omega)$, (b) $\pi_{\omega|X}(X_9, \omega)$, (c) $\pi_{\omega|X}(X_{93}, \omega)$; and (d) $\pi_{\omega|X}(X_{1526}, \omega)$ of one of the three cross-validation sets from the Schummer ovarian cancer dataset

Table 6-17: If-then fuzzy rules identified using the proposed RSPOP employed in RNFS on one of the cross-validation sets from the Schummer ovarian cancer dataset

R ₁	IF 1483 rd dot is small AND 93 rd dot is medium AND 1526 th dot is medium
R ₃	IF 1483 rd dot is small AND 9 th dot is medium AND 93 rd dot is large AND 1526 th dot is medium
R ₅	IF 1483 rd dot is small AND 9 th dot is large AND 93 rd dot is medium AND 1526 th dot is large
R ₆	IF 1483 rd dot is large AND 9 th dot is large AND 93 rd dot is large AND 1526 th dot is medium
	THEN patient is cancerous
R ₂	IF 1483 rd dot is large AND 9 th dot is medium AND 93 rd dot is large
R ₄	IF 1483 rd dot is small AND 9 th dot is medium AND 93 rd dot is large AND 1526 th dot is
	THEN patient is normal

The results show that the membership functions that characterize the possibility distributions in Figure 6-11 resemble human interpretable linguistic terms and the if-then fuzzy classification rules in Table 6-17 are humanly interpretable. Therefore, the results show that RNFS yielded a model with higher interpretability than NBPW.

Table 6-18 shows the experimental results on the confusion matrix of using RNFS that employed MIRS on the 3-fold cross validation sets.

Table 6-18: Confusion matrix on the validation sets of the Schummer ovarian cancer dataset using RNFS without rejection

Observation	CV1		CV2		CV3		Total
	Prediction		Prediction		Prediction		
	Negative	Positive	Negative	Positive	Negative	Positive	
Negative	8	0	8	0	8	0	24
Positive	0	10	0	10	0	10	30

The results in Table 6-18 show that among all the patients, none is misclassified. Therefore, rejecting the classifications with low confidence is unnecessary

In the next part of the experiment, the PrCM of NBPW is used to reject the classifications with low confidence to improve the classification accuracy. Since the results from RNFS yielded 100% test accuracy, no further improvements are required. Figure 6-17 shows the plot of the objective function $J=4e^2+r^2$ against the PrCM of NBPW. Table 6-19 shows the experimental results on the confusion matrix of using NBPW with optimal rejection using PrCM on the 3-fold cross-validation sets.

Figure 6-17: Plot of the error rate e and rejection rate r against the PrCM $CM(X)$ of NBPW on the 3-fold cross-validation sets of the Schummer ovarian cancer dataset using a objective function of $J=4e^2+r^2$

Table 6-19: Confusion matrix on the cross-validation sets of the Schummer ovarian cancer dataset using the NBPW with reject using PrCM

		CV1			CV2			CV3			
		Prediction			Prediction			Prediction			
Observation		Negative	Positive	Reject	Negative	Positive	Reject	Negative	Positive	Reject	Total
NBPW with PrCM	Negative	8	0	0	7	0	1	8	0	0	24
Threshold 34.6%	Positive	0	9	1	0	9	1	0	7	3	30

Table 6-20: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM on the 3-fold cross-validation sets of the Schummer ovarian cancer dataset

Classifier	Without rejection	With rejection	$J=4e^2+r^2$
	Accuracy (%)	CM Threshold (%)	Accuracy (%)
NBPW	92.59	34.6%	100
RNFS	100	N.A.	N.A.

Table 6-20 shows the consolidated results on the accuracy of using NBPW and RNFS with rejection versus with optimal rejections on the 3-fold cross-validation sets of the Schummer ovarian cancer dataset. Although the classification accuracy of the NBPW is significantly low without rejections, the results show that the classification accuracy improved to 100% when classification with low confidence are rejected using the objective function $J=4e^2+r^2$. The classification accuracy of the RNFS is 100% without rejections, and thus no further improvement is required. Although both NBPW and RNFS yielded significantly good accuracies, most importantly, the latter yielded a model with higher interpretability.

6.3.3 Experiment: Golub Leukemia dataset

In this section, the Golub Leukemia microarray gene expression dataset (Golub et al., 1999) is used, which consists of a total of 72 tissue samples with 7129 features where 47 samples are from patients with acute myeloid leukemia (AML) and 25 samples are from patients with acute lymphoblastic leukemia (ALL).

6.3.3.1 Feature selection and classification

Table 6-21 presents the consolidated experimental results of the training and test accuracies performed using 3-fold and 10-fold cross-validations without selection bias for the first part of the experiments. The average accuracies of using the selection of classifiers and feature selection algorithms are presented. Figure 6-18 shows a plot of the test classification accuracies against the selection of classifiers and feature selection algorithms.

Table 6-21: Consolidated experimental results of the classification accuracies of using a selection of feature selection algorithms and a selection of classifiers performed using 3-fold and 10-fold cross validations without selection bias and without rejection on the Golub Leukemia dataset

Validation Method				3-fold cross-validation		10-fold cross-validation			
Feature Selection	Features	Exec (s)	Classifier	Training Acc(%)	Test Acc(%)	Features	Training Acc(%)	Test Acc(%)	
Filter Approach	MIBIFPW	k = 3	27.2 ± 0.1	NBPW	87.5 ± 9.1	80.6 ± 13.4	k = 3.00	94.4 ± 1.1	94.3 ± 10.0
				FLD	86.8 ± 8.4	81.9 ± 12.0		90.7 ± 3.2	87.5 ± 10.5
				CART	95.1 ± 1.2	84.7 ± 12.7		96.4 ± 1.5	88.8 ± 11.2
				3-NN	95.1 ± 2.4	84.7 ± 2.4		94.8 ± 1.3	88.8 ± 11.2
				SVM	91.0 ± 4.8	87.5 ± 8.3		94.4 ± 1.1	94.3 ± 10.0
				RNFS	92.4 ± 4.3	88.9 ± 8.7		94.4 ± 1.1	94.3 ± 10.0
		DENFIS	91.0 ± 4.8	88.9 ± 6.4	94.3 ± 1.0	92.9 ± 12.1			
		k = 6	27.2 ± 0.1	NBPW	91.7 ± 7.2	87.5 ± 0.0	k = 6	90.4 ± 4.4	91.8 ± 9.8
				FLD	91.7 ± 7.2	87.5 ± 0.0		90.3 ± 4.3	91.8 ± 9.8
				CART	98.6 ± 1.2	88.9 ± 2.4		97.4 ± 1.0	88.6 ± 11.3
				3-NN	97.9 ± 2.1	90.3 ± 2.4		96.1 ± 1.5	90.0 ± 11.8
				SVM	95.8 ± 0.0	88.9 ± 2.4		95.1 ± 1.2	92.9 ± 12.1
	RNFS			97.9 ± 0.0	91.7 ± 4.2	96.1 ± 1.8		88.8 ± 13.1	
	DENFIS	98.6 ± 2.4	87.5 ± 0.0	95.5 ± 1.7	88.6 ± 13.1				
	MIPW	k = 9	27.2 ± 0.0	NBPW	93.1 ± 6.7	88.9 ± 2.4	k = 9	94.3 ± 4.4	87.3 ± 12.5
				FLD	93.1 ± 6.7	87.5 ± 4.2		97.4 ± 3.3	88.6 ± 13.1
				CART	99.3 ± 1.2	90.3 ± 2.4		99.2 ± 1.1	88.6 ± 11.3
				3-NN	98.6 ± 1.2	91.7 ± 4.2		98.0 ± 1.5	91.4 ± 13.8
				SVM	97.9 ± 0.0	93.1 ± 6.4		98.0 ± 1.5	91.4 ± 12.0
				RNFS	100.0 ± 0.0	91.7 ± 4.2		98.1 ± 1.4	91.4 ± 12.0
		DENFIS	100.0 ± 0.0	88.9 ± 2.4	98.3 ± 1.8	94.3 ± 10.0			
		k = 12	27.2 ± 0.1	NBPW	97.2 ± 3.2	93.1 ± 4.8	k = 12	96.1 ± 2.1	93.0 ± 10.0
				FLD	96.5 ± 4.3	95.8 ± 4.2		96.4 ± 2.1	93.0 ± 10.0
				CART	100.0 ± 0.0	83.3 ± 11.0		99.7 ± 0.7	87.3 ± 8.1
3-NN				97.9 ± 2.1	93.1 ± 6.4	98.9 ± 1.6		92.9 ± 10.1	
SVM				99.3 ± 1.2	93.1 ± 4.8	99.4 ± 0.8		95.7 ± 6.9	
RNFS	100.0 ± 0.0			94.4 ± 2.4	99.8 ± 0.5	94.3 ± 10.0			
DENFIS	99.3 ± 1.2	88.9 ± 6.4	99.5 ± 0.7	92.9 ± 7.5					
FRFS2	3.0 ± 0.6	169 ± 20	NBPW	99.3 ± 1.2	91.7 ± 4.2	3.0 ± 0.0	97.5 ± 3.0	82.1 ± 18.7	
			FLD	95.8 ± 2.1	91.7 ± 4.2		96.4 ± 3.3	80.7 ± 17.8	
			CART	98.6 ± 2.4	87.5 ± 11.0		98.6 ± 1.7	86.3 ± 11.2	
			3-NN	99.3 ± 1.2	91.7 ± 4.2		98.1 ± 2.2	80.7 ± 19.0	
			SVM	99.3 ± 1.2	93.1 ± 4.8		98.5 ± 1.9	86.1 ± 15.1	
			RNFS	86.8 ± 17.5	81.9 ± 14.6		77.6 ± 16.0	82.1 ± 13.0	
	DENFIS	100.0 ± 0.0	88.9 ± 6.4	99.8 ± 0.5	82.0 ± 19.0				
	3.0 ± 0.0	4.6 ± 0.3	NBPW	97.2 ± 4.8	81.9 ± 6.4	3.4 ± 0.5	98.6 ± 1.4	88.9 ± 8.9	
			FLD	95.8 ± 2.1	83.3 ± 11.0		97.7 ± 1.5	90.2 ± 6.8	
			CART	97.2 ± 2.4	77.8 ± 8.7		98.6 ± 0.9	92.9 ± 12.1	
			3-NN	97.2 ± 3.2	84.7 ± 8.7		98.0 ± 1.5	86.1 ± 9.6	
			SVM	97.2 ± 4.8	83.3 ± 7.2		98.0 ± 2.1	91.6 ± 7.2	
RNFS			65.3 ± 1.2	80.6 ± 6.4	71.6 ± 13.8		86.1 ± 9.0		
DENFIS	99.3 ± 1.2	81.9 ± 9.6	99.7 ± 0.6	85.9 ± 9.5					
Wrapper Approach	MINBPW	1.0 ± 0.0	32.2 ± 1.7	NBPW	91.0 ± 7.3	80.6 ± 14.6	1.0 ± 0.0	92.0 ± 4.9	81.8 ± 9.8
MIRSR	8.3 ± 1.5	22.0 ± 0.0	RNFS	100.0 ± 0.0	97.2 ± 2.4	9.5 ± 1.6	100.0 ± 0.0	97.1 ± 6.6	

The results show that RNFS that employed MIRSR using the wrapper approach yielded a superior test accuracy of 97.2±2.4% on 3-fold and 10-fold cross-validations.

Comparing the results in Table 6-21 with Table 6-15 and Table 6-9, superior test accuracies are again obtained from different classifiers employing different feature selection algorithm on different experiment setup. Therefore, in order to draw a more conclusive result, a consolidated result on all three datasets using different experiment setup is required. The consolidated result is presented in Section 6.3.4.

(a)

(b)

Figure 6-18: Consolidated results on the classification accuracies performed using (a) 3-fold; and (b) 10-fold cross-validations without selection bias and without rejection on the Golub Leukemia cancer dataset

6.3.3.2 Estimation of classification confidence and rejection

Table 6-22 shows the experimental results on the confusion matrix of using NBPW that employed MINBPW on 3-fold cross-validation sets for the second part of the experiments. The MIBNPW selected 3 of the 7129 features in the dataset, namely the 760th feature on gene “CYSTATIN A”, the 312nd feature on gene “NADPH-flavin

reductase” and the 1834th feature on gene “CD33 CD33 antigen”.

Table 6-22: Confusion matrix on the 3-fold validation sets of the Golub Leukemia dataset using NBPW without rejection

Observation	CV1		CV2		CV3		Total
	Prediction		Prediction		Prediction		
	Negative	Positive	Negative	Positive	Negative	Positive	
Negative	16	0	16	0	15	0	47
Positive	2	6	2	6	0	9	25

Table 6-22 shows that among the 74 patients with acute Leukemia cancer, 4 are misclassified.

Figure 6-19 shows the surface plot of the joint posterior probability and PrCM of NBPW against the FLD transformed features on one of the three cross-validation sets.

Figure 6-19: Plot of the joint probability (a) $p(X|\omega_1)$ for $\omega=1$, (b) $p(X|\omega_2)$ $\omega=2$, (c) $p(X|\omega)$ for both $\omega=1$ and 2 and (d) plot of the PrCM $CM(X)$ of NBPW against FLD transformed features t_1 and t_2 of one of the three cross-validation sets from the Golub Leukemia dataset

Figure 6-19(d) shows that the samples from patients with AML ($\omega=1$) and ALL ($\omega=2$) are well separated with a complex decision boundary. The results show that there are a number of samples from both classes at the decision boundary. Hence, the width of the uncertainty region at the decision boundary is significantly large. Again, this result shows that the PrCM is suitable for estimating the classification confidence of modeling the Golub Leukemia dataset.

Figure 6-20: Plot of the membership functions generated using the proposed SPSEC employed in RNFS that characterize the possibility distributions (a) $\pi_{\omega|\chi}(X_{6885}, \omega)$; (b) $\pi_{\omega|\chi}(X_{1685}, \omega)$; (c) $\pi_{\omega|\chi}(X_{758}, \omega)$; (d) $\pi_{\omega|\chi}(X_{1114}, \omega)$; (e) $\pi_{\omega|\chi}(X_{2354}, \omega)$; (f) $\pi_{\omega|\chi}(X_{4680}, \omega)$; and (g) $\pi_{\omega|\chi}(X_{2642}, \omega)$ of one of the three cross-validation sets from the Golub Leukemia dataset

Next, the experiment is repeated using RNFS. Figure 6-20 shows the membership functions generated using the proposed SPSEC employed in RNFS, and the proposed MIRSR employed in RNFS selected 11 out of the 7129 features in the dataset. These

11 features are further reduced to 7 features by the proposed RSPOP employed in RNFS; namely, the 6855th feature on gene “TCF3”, the 1685th feature on gene “Terminal transferase mRNA”, the 758th feature on gene “GB DEF”, the 1144th feature on gene “SPTAN1” the 2354th feature on gene “CCND3”, the 4680th feature on gene “TCL1”, and the 2642nd feature on gene “MB-1”. Table 6-23 shows the if-then fuzzy rules identified using the proposed RSPOP employed in RNFS.

Table 6-23: If-then fuzzy rules identified using the proposed RSPOP employed in RNFS on one of the cross-validation sets from the Golub Leukemia dataset

R_1	IF TCF3 is large AND Terminal transferase mRNA is large AND SPTAN1 is large AND CCND3 Cyclin D3 is large AND TCL1 is large AND MB-1 is large
R_3	IF TCF3 is large AND GB DEF is medium AND SPTAN1 is large AND CCND3 Cyclin D3 is large AND TCL1 is medium AND MB-1 is medium
R_4	IF TCF3 is large AND Terminal transferase mRNA is large AND GB DEF is large AND SPTAN1 is large AND TCL1 is medium AND MB-1 is large
R_5	IF TCF3 is large AND Terminal transferase mRNA is large AND GB DEF is large AND CCND3 Cyclin D3 is medium AND TCL1 is large AND MB-1 is large
R_6	IF TCF3 is medium AND Terminal transferase mRNA is large AND GB DEF is large AND SPTAN1 is large AND CCND3 Cyclin D3 is large AND TCL1 is medium AND MB-1 is medium
R_7	IF TCF3 is medium AND Terminal transferase mRNA is medium AND GB DEF is large AND SPTAN1 is medium AND CCND3 Cyclin D3 is large AND TCL1 is large AND MB-1 is large THEN patient is AML
R_2	IF TCF3 is medium AND Terminal transferase mRNA is medium AND GB DEF is medium AND SPTAN1 is medium AND CCND3 Cyclin D3 is medium AND TCL1 is medium AND MB-1 is medium THEN patient is ALL

The results show that the membership functions that characterize the possibility distributions in Figure 6-20 resemble human interpretable linguistic terms and the if-then fuzzy classification rules in Table 6-23 are humanly interpretable. Therefore, the results show that RNFS yielded a model with higher interpretability than NBPW. Table 6-24 shows the experimental results on the confusion matrix of using RNFS that employed MIRS on the 3-fold cross-validation sets.

Table 6-24: Confusion matrix on the 3-fold validation sets of the Golub Leukemia dataset using NBPW without rejection

Observation	CV1		CV2		CV3		Total
	Prediction		Prediction		Prediction		
	Negative	Positive	Negative	Positive	Negative	Positive	
Negative	16	0	15	1	15	0	47
Positive	0	8	0	8	1	8	25

Table 6-24 shows that among the 74 patients with acute Leukemia cancer, 2 are misclassified.

Figure 6-21 shows the surface plot of the joint possibility and PoCM of RNFS against the FLD transformed features on one of the three cross-validation sets.

Figure 6-21: Plot of the joint possibility distribution (a) $\pi_{\omega_1}(X, \omega_1)$ for $\omega=1$, (b) $\pi_{\omega_2}(X, \omega_2)$ for $\omega=2$, (c) $\pi_{\omega}(X, \omega)$ for both $\omega=1$ and 2 and (d) plot of the PoCM $CM(X)$ of RNFS against transformed features t_1 and t_2 of t_1 and t_2 of one of the cross-validation sets from the Golub Leukemia dataset

Figure 6-21(d) shows that the samples from patients with AML ($\omega=1$) and ALL ($\omega=2$) are not very well separated, and a complex decision boundary is generated. The results show that there are a small number of samples from both classes at the complex decision boundary, but the width of the uncertainty region at the complex decision boundary is small. This result shows that the proposed PoCM is suitable for estimating the classification confidence of modeling the Golub Leukemia dataset.

In the next part of the experiment, the PrCM of NBPW and PoCM of RNFS are used to reject the classifications with low confidence to improve the classification accuracies. Figures 6-22(a) and (b) show the plot of the objective function $J=4e^2+r^2$ against the PrCM of NBPW and PoCM of RNFS respectively. The results show that the former yielded an optimal rejection PrCM threshold of 61.8% and the latter yielded an optimal rejection PoCM threshold of 65.2%. Table 6-25 shows the experimental results on the confusion matrix of using NBPW and RNFS with optimal rejection using PrCM and PoCM on the 3-fold cross-validation sets.

Figure 6-22: Plot of the error rate e and rejection rate r against (a) PrCM $CM(X)$ of NBPW and (b) PoCM $CM(X)$ of RNFS on the 3-fold cross-validation sets of the Golub Leukemia dataset using an objective function of $J=4e^2+r^2$

Table 6-25: Confusion matrix on the cross-validation sets of the Golub Leukemia dataset using NBPW and RNFS with optimal rejection using PrCM and PoCM respectively

		CV1			CV2			CV3			
		Prediction			Prediction			Prediction			
Observation		Negative	Positive	Reject	Negative	Positive	Reject	Negative	Positive	Reject	Total
NBPW with PrCM	Negative	16	0	0	16	0	0	15	0	0	47
Threshold 61.8%	Positive	0	6	2	0	6	2	0	5	4	25
RNFS with PoCM	Negative	16	0	0	14	0	2	15	0	0	17
Threshold 65.2%	Positive	0	7	1	0	8	0	0	7	2	139

Table 6-26: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the Golub Leukemia dataset

Classifier	Without rejection	With rejection	$J=4e^2+r^2$
	Accuracy (%)	CM Threshold (%)	Accuracy (%)
NBPW	94.44	61.8	100
RNFS	97.22	65.2	100

Table 6-26 shows the consolidated results on the accuracy of using NBPW and RNFS without rejections versus with optimal rejection using PrCM and PoCM respectively

on the 3-fold cross-validation sets of the Golub Leukemia dataset. Although the classification accuracies are significantly high without rejections, the results show that the classification accuracies improved further when classification with low confidence are rejected using the objective function $J=4e^2+r^2$. Thus, the results again show that the use of PrCM and PoCM to perform optimal rejection improved the classification accuracies of NBPW and RNFS respectively.

6.3.4 Consolidated experimental results

This section consolidates the results from sections 6.3.1 to 6.3.3 on the three datasets.

6.3.4.1 Feature selection and classification

Table 6-27 presents the consolidated experimental results of the training and test accuracies from the three datasets performed using 3-fold and 10-fold cross-validations without selection bias for the first part of the experiments. The average test accuracies from 3-fold and 10-fold cross-validations of using the selection of classifiers and feature selection algorithms are also presented. Figure 6-23 shows a plot of the averaged test classification accuracies from the three datasets against the selection of classifiers and feature selection algorithms.

Table 6-27: Consolidated experimental results of the classification accuracies from the three datasets using a selection of feature selection algorithms and a selection of classifiers performed using 3-fold and 10-fold cross validations without selection bias and without rejection

Validation Method				3-fold cross-validation		10-fold cross-validation		Mean for 3 & 10-fold cross-validation				
Feature Selection	Features	Exec (s)	Classifier	Training Acc(%)	Test Acc(%)	Features	Training Acc(%)	Test Acc(%)	Training Acc(%)	Test Acc(%)		
Filter Approach	MIBIFPW	k = 3	4.9 ± 0.0	NBPW	84.0 ± 11.8	80.7 ± 12.9		84.3 ± 5.1	83.2 ± 7.9	84.2 ± 8.4	82.0 ± 10.4	
				FLD	78.0 ± 16.8	73.5 ± 19.4		81.1 ± 8.8	75.8 ± 12.3	79.5 ± 12.8	74.7 ± 15.9	
				CART	93.9 ± 4.5	78.3 ± 13.2		93.2 ± 3.1	80.4 ± 11.0	93.6 ± 3.8	79.4 ± 12.1	
				3-NN	89.7 ± 9.5	83.4 ± 8.4	k = 3	91.9 ± 4.5	84.1 ± 15.5	90.8 ± 7.0	83.8 ± 11.9	
				SVM	85.2 ± 10.1	83.0 ± 9.7		84.5 ± 4.8	83.8 ± 7.8	84.8 ± 7.5	83.4 ± 8.8	
				RNFS	71.7 ± 21.2	83.6 ± 8.6		83.2 ± 12.5	83.0 ± 8.8	77.4 ± 16.8	83.3 ± 8.7	
				DENFIS	83.9 ± 8.7	82.4 ± 9.0		86.0 ± 4.2	83.3 ± 9.9	84.9 ± 6.5	82.9 ± 9.4	
		k = 6	4.9 ± 0.0	NBPW	90.5 ± 11.3	87.3 ± 8.8		91.4 ± 6.7	89.3 ± 9.8	91.0 ± 9.0	88.3 ± 9.3	
				FLD	86.8 ± 16.8	82.8 ± 17.8		89.7 ± 9.5	87.6 ± 13.9	88.3 ± 13.1	85.2 ± 15.9	
				CART	97.6 ± 3.0	86.5 ± 7.3		97.1 ± 2.0	87.9 ± 12.1	97.4 ± 2.5	87.2 ± 9.7	
				3-NN	95.3 ± 6.2	86.6 ± 10.3	k = 6	95.7 ± 2.6	89.4 ± 13.4	95.5 ± 4.4	88.0 ± 11.9	
				SVM	92.6 ± 8.6	88.2 ± 9.4		93.1 ± 5.5	90.2 ± 12.2	92.9 ± 7.0	89.2 ± 10.8	
				RNFS	94.5 ± 7.2	91.0 ± 8.6		94.8 ± 5.8	90.7 ± 10.1	94.7 ± 6.5	90.9 ± 9.4	
				DENFIS	94.6 ± 8.0	88.1 ± 2.8		92.7 ± 6.3	88.4 ± 11.5	93.7 ± 7.2	88.3 ± 7.2	
	MIPW	k = 9	4.9 ± 0.0	NBPW	93.7 ± 4.8	90.7 ± 5.1		93.9 ± 3.2	90.9 ± 10.0	93.8 ± 4.0	90.8 ± 7.6	
				FLD	91.1 ± 8.1	86.6 ± 10.1		94.1 ± 5.3	90.5 ± 12.9	92.6 ± 6.7	88.5 ± 11.5	
				CART	98.7 ± 2.2	90.5 ± 2.5		98.2 ± 1.6	92.2 ± 9.7	98.5 ± 1.9	91.3 ± 6.1	
				3-NN	97.6 ± 3.1	92.2 ± 2.0	k = 9	97.8 ± 1.7	94.5 ± 9.8	97.7 ± 2.4	93.4 ± 5.9	
				SVM	96.9 ± 3.6	94.1 ± 5.3		97.4 ± 2.3	93.6 ± 9.0	97.1 ± 3.0	93.9 ± 7.2	
				RNFS	98.1 ± 3.3	94.3 ± 4.1		98.2 ± 2.0	93.4 ± 9.0	98.1 ± 2.7	93.8 ± 6.5	
				DENFIS	97.1 ± 4.1	87.3 ± 8.8		97.6 ± 2.5	94.4 ± 8.8	97.3 ± 3.3	90.9 ± 8.8	
		k = 12	4.8 ± 0.0	NBPW	95.3 ± 2.7	91.4 ± 5.0		94.9 ± 2.1	91.7 ± 12.0	95.1 ± 2.4	91.6 ± 8.5	
				FLD	93.0 ± 6.0	90.6 ± 8.2		93.8 ± 4.6	90.5 ± 14.6	93.4 ± 5.3	90.5 ± 11.4	
				CART	99.4 ± 1.1	87.3 ± 6.4		98.8 ± 1.2	92.1 ± 8.1	99.1 ± 1.1	89.7 ± 7.2	
				3-NN	97.2 ± 3.0	92.3 ± 5.3	k = 12	98.3 ± 1.2	94.9 ± 8.9	97.8 ± 2.1	93.6 ± 7.1	
				SVM	98.5 ± 2.3	94.6 ± 4.1		98.2 ± 1.9	94.5 ± 8.3	98.4 ± 2.1	94.5 ± 6.2	
				RNFS	98.6 ± 2.3	94.2 ± 4.9		99.1 ± 1.3	94.2 ± 8.5	98.9 ± 1.8	94.2 ± 6.7	
				DENFIS	99.2 ± 1.3	86.5 ± 7.9		98.4 ± 1.9	90.5 ± 11.0	98.8 ± 1.6	88.5 ± 9.4	
	MIPW	2.7 ± 0.4	24.1 ± 1.8	NBPW	98.9 ± 1.8	93.9 ± 2.8		97.4 ± 2.1	90.0 ± 12.2	98.2 ± 2.0	91.9 ± 7.5	
				FLD	97.2 ± 2.5	93.1 ± 3.2		95.7 ± 2.7	86.9 ± 13.3	96.4 ± 2.6	90.0 ± 8.2	
				CART	98.9 ± 1.3	92.1 ± 6.0		98.0 ± 1.6	90.8 ± 9.5	98.4 ± 1.5	91.4 ± 7.8	
				3-NN	98.4 ± 1.5	93.5 ± 4.2	2.7 ± 0.1	97.5 ± 1.6	88.3 ± 13.2	98.0 ± 1.6	90.9 ± 8.7	
				SVM	98.6 ± 1.7	94.3 ± 4.4		96.9 ± 1.8	90.2 ± 11.5	97.8 ± 1.7	92.3 ± 8.0	
				RNFS	64.7 ± 7.1	88.6 ± 8.2		67.0 ± 18.5	89.1 ± 11.6	65.8 ± 12.8	88.8 ± 9.9	
				DENFIS	100.0 ± 0.0	93.2 ± 4.6		99.4 ± 0.8	89.5 ± 12.5	99.7 ± 0.4	91.3 ± 8.5	
		FRFS2	2.7 ± 0.4	1.5 ± 0.2	NBPW	98.2 ± 3.0	89.4 ± 4.3		97.6 ± 1.6	92.4 ± 9.0	97.9 ± 2.3	90.9 ± 6.7
				FLD	96.7 ± 2.1	89.3 ± 6.2		96.2 ± 2.0	92.2 ± 8.2	96.4 ± 2.1	90.7 ± 7.2	
				CART	98.8 ± 1.3	88.0 ± 5.3		98.3 ± 1.2	93.7 ± 9.6	98.6 ± 1.3	90.8 ± 7.4	
				3-NN	97.9 ± 1.8	88.9 ± 6.1	3.1 ± 0.3	97.6 ± 1.7	90.2 ± 10.3	97.8 ± 1.7	89.5 ± 8.2	
				SVM	97.4 ± 3.9	89.7 ± 4.6		97.1 ± 1.9	93.0 ± 7.9	97.3 ± 2.9	91.4 ± 6.3	
				RNFS	61.5 ± 27.5	85.8 ± 5.9		62.5 ± 22.1	91.6 ± 8.9	62.0 ± 24.8	88.7 ± 7.4	
				DENFIS	99.5 ± 0.9	88.0 ± 6.4		99.3 ± 0.8	89.8 ± 9.9	99.4 ± 0.9	88.9 ± 8.1	
Wrapper Approach	MINBPW	1.8 ± 0.2	6.4 ± 0.3	NBPW	93.5 ± 5.0	88.3 ± 8.8	1.9 ± 0.2	95.0 ± 2.7	87.6 ± 8.9	94.3 ± 3.9	88.0 ± 8.8	
	MIRSR	7.4 ± 2.0	7.9 ± 0.2	RNFS	100.0 ± 0.0	94.7 ± 1.9	8.6 ± 2.1	98.2 ± 5.8	95.9 ± 6.4	99.1 ± 2.9	95.3 ± 4.1	

The results in tables 6-9, 6-15 and 6-21 show that the RNFS that employed MIRSR yield superior test accuracies on certain parts of three experiments performed; namely, 10-fold cross-validations of the Bhattacharjee lung cancer dataset, 3-fold and 10-fold cross-validations of the Golub leukemia dataset. This is discussed in section 6.3.2.1 as the *No Free Lunch Theorem* in pattern recognition literature (Duda et al., 2001). Nevertheless, the consolidated results in Table 6-27 from the three datasets show that

the RNFS that employed MIRSRS using the wrapper approach yielded a superior averaged test accuracy of $95.3 \pm 4.1\%$ from 3-fold and 10-fold cross-validations of the three datasets.

Figure 6-23: Averaged results on the classification accuracies from the three datasets performed using 3-fold and 10-fold cross-validations without selection bias and without rejection

6.3.4.2 Estimation of classification confidence and rejection

Table 6-28 presents the consolidated results on the accuracy of using NBPW and RNFS without rejections versus with optimal rejection using PrCM and PoCM respectively on the 3-fold cross-validation of the three datasets.

Table 6-28: Consolidated results on the accuracy of NBPW and RNFS without optimal rejections versus with optimal rejections using PrCM and PoCM respectively on the 3-fold cross-validation sets of the three datasets

Dataset	Classifier	Without rejection	With rejection
		Accuracy (%)	Accuracy (%)
Bhattacharjee	NBPW	98.08	100
lung cancer	RNFS	96.15	99.35
Schummer	NBPW	92.59	100
ovarian cancer	RNFS	100	N.A.
Golub	NBPW	94.44	100
Leukemia	RNFS	97.22	100

The results show that the classification accuracies of both the NBPW and RNFS can be improved by rejecting classifications with low confidence using PrCM and PoCM respectively on all three datasets. The results also show that the RNFS yielded 100%

test accuracy without rejecting classifications with low confidence. Thus, rejection of classification with low confidence using the RNFS on the Schummer Ovarian Cancer dataset is unnecessary. This result shows that for real applications, a cross-validation on available data can be used to determine if rejection of classifications with low confidence is needed. If the cross-validation yielded a high accuracy of 100%, rejection of classifications with low confidence is unnecessary.

6.4 Summary

This Chapter demonstrates the effectiveness of using RNFS in this biomedical case study of modeling classification data on samples from cancerous and normal patients. In the first set of experiments, three-dimensional visualization of the proposed PrCM of NBPW and PoCM of RNFS are presented using the Pima Indians diabetes dataset. This result reinforced the studies presented in section 4.5 to show that they are able to estimate the classification confidence so that classifications with low confidence can be rejected. The results show that the classification accuracies of NBPW and RNFS are significantly improved when classifications with low confidence are rejected using an objective function to compute the optimal confidence threshold.

In the second set of experiments performed without selection bias on higher dimensional datasets, the results showed that RNFS that employed MIRSR without rejection yielded superior averaged test accuracy compared against the selection of classifiers and feature selection algorithms on three datasets. The results also show that classification accuracies of RNFS can be further improved by rejecting classifications with low confidence using PoCM. In addition, the results show that RNFS yielded a model with higher interpretability than NBPW.

The potential of classifying biomedical data using RNFS discussed in this chapter is

encouraging as it demonstrates the feasible application of RNFS in the area of biomedical computation.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

This thesis focused on abstracting a fuzzy model using neural learning techniques to produce articulated knowledge in a human interpretable form. Prevailing research in neuro-fuzzy systems focused on increasing accuracy while neglecting interpretability. In this thesis, a novel hybrid intelligent Rough set-based Neuro-Fuzzy System (RNFS) is proposed to improve the interpretability without compromising the accuracy of the abstracted model. Figure 7-1 shows a consolidated view of the contributions of RNFS in this thesis. Table 7-1 gives a summary of the research achievements with respect to each of the contributions.

Figure 7-1: A consolidated view of the contributions of RNFS in this thesis

Table 7-1: A summary of research achievements in this thesis

Part	Contribution	Proposed in	Problems addressed	Compared in	Applied in
(a)	SPSEC	Section 3.3.1.1	Generation of effective and interpretable membership functions	Section 4.2	(i) Modeling of regression data in the financial case study of trading and predicting stock price in Chapter 5
(b)	RSPOP	Section 3.3.2.1	Identification of effective and interpretable if-then fuzzy rules	Section 4.3	
(c)	MIRSR	Section 3.4.1	Selection of features to model high dimensional classification data	Section 4.4	(ii) Modeling of high dimensional classification data in the biomedical case study of classifying microarray gene expression dataset in Chapter 6
(d)	PoCM	Section 3.5.2	Rejection of prediction with low confidence to increase accuracy	Section 4.5	

In part (a), the problem of generating effective and interpretable membership is addressed with the Supervised Pseudo Self-Evolving Cerebellar (SPSEC) algorithm proposed in section 3.3.1.1. SPSEC employs a novel two-stage learning technique that first constructs a cerebellar-like structure in which neurons with high trophic factors evolve to form the membership functions. The comparative study in section 4.2 has shown that SPSEC is capable of generating membership functions that conform to the semantic properties of interpretable linguistic variables and inherent probabilistic decision boundaries of the training data without requiring user-defined parameters.

In part (b), the problem of identifying effective and interpretable if-then fuzzy rules is addressed with the Rough Set-based Pseudo Outer-Product (RSPOP) algorithm proposed in section 3.3.2.1. RSPOP synergizes the theory of knowledge reduction in rough set theory with the Hebbian learning technique to identify non-redundant if-then fuzzy rules without requiring user-defined parameters. The comparative study in section 4.3 has shown that the RSPOP is capable of identifying a small number of humanly interpretable if-then fuzzy rules, which improve the accuracy of the abstracted model to the extent that it is comparable to precise neuro-fuzzy system.

An application of parts (a) and (b) of the RNFS in modeling low dimensional regression data is presented on a financial case study in Chapter 5. The effectiveness of the RNFS in the modeling and prediction of real world stock price is measured in terms of trading profits that is more important to an investor than statistical performance of the prediction. The results have shown that the stock trading model using RNFS identifies smaller number of if-then fuzzy rules with greater interpretability and yields significantly higher profits compared to a stock trading model using precise neuro-fuzzy system.

In part (c), the problem of selecting features in modeling high dimensional classification data is addressed with the Mutual Information-based Rough Set Reduction (MIRSR) feature selection algorithm using the wrapper approach proposed in section 3.4.1. MIRSR synergizes the selection of relevant features with the reduction of redundant features. The former employs the MI computed using the membership functions generated from SPSEC while the latter employs RSPOP attribute reduction. The comparative study in section 4.4 has shown that the RNFS that employs MIRSR yields superior cross-validation accuracy compared against a selection of classification and feature selection algorithms.

In part (d), the problem of rejecting the classification with low confidence is addressed with the Possibilistic Confidence Measure (PoCM) proposed in section 3.5.2. The PoCM is computed using the compositional rule of inference and the union of the if-then fuzzy rules of the RNFS. The comparative study in section 4.5 has shown that the PoCM is capable of estimating the classification confidence.

An application of parts (a) to (d) of the RNFS in modeling high dimensional classification data is presented in the biomedical case study in Chapter 6. The effectiveness of the RNFS in the modeling and classification of real world cancerous microarray gene expressions is measured in terms of the accuracy of classification, which is the most important performance measure to a doctor or a patient. The results have shown that the RNFS that employs MIRSR yields superior averaged cross-validation accuracy from three datasets compared against a selection of classification and feature selection algorithms. The results have also shown that the PoCM is capable of increasing the accuracy of the RNFS by evaluating the reliability of the classification so that the sample with low classification confidence can be rejected rather than risking a wrong classification.

7.2 Recommendations for future work

Although the comparative studies and the case studies have shown that the proposed RNFS in this thesis yields superior results in terms of interpretability and accuracy compared against selected linguistic neuro-fuzzy systems, further research can be pursued in the following aspects:

- *Default parameters* – It has been shown that RNFS does not require any user-defined parameters. However, the proposed algorithms employed in RNFS utilized parameters with default values in order to abstract a model autonomously from numerical training data. The parameters with default values utilized are namely, the neighborhood h_ω in equation (3.1) utilized by SPSEC, the number of output membership functions N_ω utilized by SPSEC for modeling regression data, the initial number of neurons per class m utilized by SPSEC, and the initial selection of k best features before the removal of redundant features in equation (3.31) utilized by MIRS. The default value for h_ω is based on the normal optimal smoothing (Bowman & Azzalini, 1997) and thus is not investigated further. The default value for k in equation (3.31) is arbitrarily determined from the number of total features since a selection of a large k may result in a lengthier computation by MIRS to remove additional redundant features. Although the number of N_ω is known in the modeling of classification data, the value of this parameter affects the number of if-then fuzzy rules identified and the accuracy of the abstracted model in modeling regression data. Therefore, further studies on the default values for these parameters should be carried out.
- *Over-trained phenomenon* – Although SPSEC has been shown to be able to generate effective membership functions for modeling classification data in section 4.2, it has been shown in section 5.3 that the RNFS that employs SPSEC yields

superior recall but inferior predictive results than the RNFS that employs manually constructed membership functions. This is because the RNFS that employs SPSEC generates a large number of membership functions to fit the artificially generated noisy price data, which results in an increased number of if-then fuzzy rules that causes the over-trained phenomenon. This issue is also known as overfitting avoidance and Occam's razor in pattern recognition literature (Duda et al., 2001). Therefore, further studies on how to avoid the overfitting the training data by SPSEC should be carried out.

- *Outlier rule reduction* – In addition to RSPOP attribute and rule reductions, a further reduction of the if-then fuzzy rules can be autonomously performed by removing the outlier rules. An outlier rule can be identified from one with the least firing strength among all the existing if-then fuzzy rules, in which the firing strength is computed from equation (3.14). Similar to RSPOP attribute and rule reduction, a deterioration of objective measure in using the outlier-reduced rules can be determined from the baseline objective measure if equation (3.20) is true. Therefore, further investigation on outlier rule reduction, which may improve the interpretability of the abstracted model, is highly desirable.
- *Compositional Rule of Inference (CRI)* – The RNFS proposed in this thesis is fundamentally based on the architecture of POPFNN-CRI(S) (Ang et al., 2003). The CRI method originally proposed by Zadeh (1973) comprises the max-min composition operator and the implication operator. Thus, the inference pattern depends on the choice of these operators. Baldwin and Pilsworth (1980) and Hall (1990) have compared various inference patterns based on various criteria and showed that the use of the min-max composition operator and the Łukasiewicz implication operator satisfied all the criteria. Hall (1990) also indicated that the

Łukasiewicz implication operator should be used when there is an absence of large amounts of uncertainty. Hence, the use of the Łukasiewicz implication operator instead of the Mamdani implication operator in RNFS is worth investigating.

- *Reasoning method* – The contributions of RNFS in Figure 7-1 involve only two out of the five fundamental components of fuzzy systems reviewed in section 2.3. The other component that is worth investigating is the *reasoning method*. Besides POPFNN-CRI(S), the POPFNN family reviewed in section 2.4.4 includes members that use other reasoning methods such as *Truth Value Restriction* method (TVR) (de Mántaras, 1990), and *Approximate Analogical Reasoning Schema* (AARS) (Turksen & Zhong, 1990). Hence, an extension of RNFS using these other reasoning methods and a comparative study of RNFS that employs CRI with RNFS that employs TVR and AARS could yield interesting results.
- *Other applications* – The potential of RNFS has been demonstrated in the modeling of traffic flow data, stock price data and microarray gene expressions. The case study of applying RNFS in modeling stock price data revealed the importance of developing an application specific performance measure, which is the trading profit in terms of portfolio end value. The potential of RNFS in the autonomously modeling of data is encouraging and thus can be further exploited in other application areas equipped with an application specific performance measure. A non-exhaustive list of challenging applications include: the modeling of electroencephalogram (EEG) signal for Brain Computer Interface (Li & Guan, 2006), modeling of artificial ventilation control for the treatment of patients who require breathing assistance in intensive care (Liu, Ng, Quek, & Loh, 2006), and modeling of effective gain-schedule for complex control systems such as Continuous Variable Transmission (CVT) (Ang, Quek, & Wahab, 2002).

References

- Adya, M., & Collopy, F. (1998). How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting*, 17(5-6), 481-495.
- Albus, J. S. (1975a). Data storage in the cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 97, 228-233.
- Albus, J. S. (1975b). A new approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC). *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 97, 270-277.
- Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12), 6745-6750.
- Ambroise, C., & McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10), 6562-6566.
- Ang, K. K. (1998). *POPFNN-CRI(S) : a fuzzy neural network based on the compositional rule of inference*. M. Phil. Thesis, School of Applied Science, Nanyang Technological University, Singapore.
- Ang, K. K., & Quek, C. (2005). RSPOP: Rough Set-Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. *Neural Computation*, 17(1), 205-243 .
- Ang, K. K., & Quek, C. (2006a). Rough Set-based Neuro-Fuzzy System. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '06)* (pp. 742-749).
- Ang, K. K., & Quek, C. (2006b). Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach. *IEEE Transactions on Neural Networks*, 17(5), 1301-1315.
- Ang, K. K., Quek, C., & Pasquier, M. (2003). POPFNN-CRI(S): pseudo outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 33(6), 838-849.
- Ang, K. K., Quek, C., & Wahab, A. (2002). MCMAC-CVT: a novel on-line associative memory based CVT transmission control system. *Neural Networks*, 15(2), 219-236.
- Babich, G. A., & Camps, O. I. (1996). Weighted Parzen windows for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5), 567-570.
- Babuška, R., & Verbruggen, H. (2003). Neuro-fuzzy methods for nonlinear system identification. *Annual Reviews in Control*, 27(1), 73-85.
- Baldwin, J. F., & Pilsworth, B. W. (1980). Axiomatic approach to implication for approximate reasoning with fuzzy logic. *Fuzzy Sets and Systems*, 3(2), 193-219.
- Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning problems. *IEEE Transactions on Systems, Man and Cybernetics*, 13(5), 834-846.
- Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5 (4), 537-550.
- Bellman, R., Kalaba, R., & Zadeh, L. A. (1966). Abstraction and pattern classification. *Journal of Mathematical Analysis Applications*, 13, 1-7.
- Bellman, R. E., & Zadeh, L. A. (1970). Decision-Making in a Fuzzy Environment. *Management Science*, 17(4, Application Series), B141-B164.
- Berenji, H. R., & Khedkar, P. (1992). Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Transactions on Neural Networks*, 3(5), 724-740.
- Berthold, M. R. (2003). Fuzzy Logic. In M. Berthold, & D. J. Hand (Eds.), *Intelligent Data Analysis* (2nd ed., pp. 321-350). Berlin, Heidelberg: Springer-Verlag.

- Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press.
- Bezdek, J. C. (1994). What is computational intelligence? In J. M. Zurada, R. J. Marks II, & C. J. Robinson (Eds.), *Computational Intelligence: Imitating Life* (pp. 1-12). New York: IEEE Press.
- Bezdek, J. C., Dubois, D., & Prade, H. (1999). *Fuzzy Sets in Approximate Reasoning and Information Systems* (The Handbooks of Fuzzy Sets Series, No. 5). Boston: Kluwer Academic Publishers.
- Bezdek, J. C., Keller, J., Krisnapuram, R., & Pal, N. R. (1999). *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing* (The Handbooks of Fuzzy Sets Series, No. 4). Boston: Kluwer Academic Publishers.
- Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P. et al. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences of the United States of America*, 98(24), 13790-13795.
- Bowman, A. W., & Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*. New York: Oxford University Press.
- Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control* (3rd ed.). Englewood Cliffs, N.J.: Prentice Hall.
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees* (The Wadsworth statistics/probability series). Belmont, CA: Wadsworth International Group.
- Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. *Journal of Finance*, 47(5), 1731-1764.
- Broomhead, D. S., & Lowe, D. (1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems*, 2(3), 321-355.
- Buckley, J. J., & Hayashi, Y. (1994). Fuzzy neural networks: A survey. *Fuzzy Sets and Systems*, 66(1), 1-13.
- Buckley, J. J., & Yoichi Hayashi. (1995). Neural nets for fuzzy systems. *Fuzzy Sets and Systems*, 71(3), 265-276.
- Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(5), 698-713.
- Casillas, J., Cordón, O., Herrera, F., & Magdalena, L. (2003). *Interpretability Issues in Fuzzy Modeling* (Studies in fuzziness and soft computing, No. 128). Berlin: Springer-Verlag.
- Castro, J. L. (1995). Fuzzy logic controllers are universal approximators. *IEEE Transactions on Systems, Man and Cybernetics*, 25(4), 629-635.
- Chakraborty, D., & Pal, N. R. (2004). A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification. *IEEE Transactions on Neural Networks*, 15(1), 110-123.
- Chang, S. S. L., & Zadeh, L. A. (1972). On Fuzzy Mapping and Control. *IEEE Transactions on Systems, Man and Cybernetics*, 2(1), 30-34.
- Chen, A.-S., Leung, M. T., & Daouk, H. (2003). Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index. *Computers & Operations Research*, 30(6), 901-923.
- Chen, G., & Pham, T. T. (2006). *Introduction to fuzzy systems*. Boca Raton, FL: Chapman & Hall/CRC.
- Chen, S., Cowan, C. F. N., & Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis. *IEEE Transactions on Neural Networks*, 2(2), 302-309.
- Chiu, S. (1996). Method and software for extracting fuzzy classification rules by subtractive clustering. *Proceedings of the North American Fuzzy Information Processing Society (NAFIPS'96)* (pp. 461-465).
- Chiu, S. L. (1994). Fuzzy Model Identification Based on Cluster Estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3), 267-278.

- Chou, P. A. (1991). Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 340-354.
- Chow, C. K. (1970). On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1), 41-46.
- Chow, T. W. S., & Huang, D. (2005). Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Transactions on Neural Networks*, 16(1), 213-224.
- Civanlar, M. R., & Trussell, H. J. (1986). Constructing membership functions using statistical data. *Fuzzy Sets and Systems*, 18(1), 1-13.
- Clements, M. P., & Hendry, D. F. (1999). *Forecasting non-stationary economic time series*. Cambridge, Mass.: MIT Press.
- Cordella, L. P., De Stefano, C., Tortorella, F., & Vento, M. (1995). A method for improving classification reliability of multilayer perceptrons. *IEEE Transactions on Neural Networks*, 6(5), 1140-1147.
- Cordon, O., & Herrera, F. (1999). A two-stage evolutionary process for designing TSK fuzzy rule-based systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 29(6), 703-715.
- Corradi, V., & Swanson, N. R. (2006). The effect of data transformation on common cycle, cointegration, and unit root tests: Monte Carlo results and a simple test. *Journal of Econometrics*, 132(1), 195-229.
- Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3), 273-297.
- Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21-27.
- Cover, T. M. (1974). The Best Two Independent Measurements Are Not the Two Best. *IEEE Transactions on Systems, Man and Cybernetics*, 4(1), 116-117.
- Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory* (2nd ed.). New York: Wiley.
- Cox, E. (1999). *The fuzzy systems handbook : a practitioner's guide to building, using, and maintaining fuzzy systems* (2nd ed.). San Diego: AP Professional.
- Curry, B. (2003). Rough sets: current and future developments. *Expert Systems*, 20(5), 247-250.
- Dasarathy, B. V. (1991). *Nearest neighbor (NN) norms : NN pattern classification techniques*. Los Alamitos, Calif. : IEEE Computer Society Press.
- de Mántaras, R. L. (1990). *Approximate Reasoning Models*. Chichester, England: Ellis Horwood.
- de Oliveira, J. V. (1999a). Semantic constraints for membership function optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 29(1), 128-138.
- de Oliveira, J. V. (1999b). Towards neuro-linguistic modeling: Constraints for optimization of membership functions. *Fuzzy Sets and Systems*, 106(3), 357-380.
- De Stefano, C., Sansone, C., & Vento, M. (2000). To reject or not to reject: that is the question-an answer in case. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 30(1), 84-94 .
- Delany, S. J., Cunningham, P., Doyle, D., & Zolotarevskikh, A. (2005). Generating estimates of classification confidence for a case-based spam filter. In H. Munoz-Avila, & F. Ricci (Eds.), *Case-Based Reasoning Research and Development - ICCBR 2005 Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)* (Vol. 3620, pp. 177-190). Berlin, Heidelberg: Springer.
- Denoeux, T., & Zouhal, L. M. (2001). Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Sets and Systems*, 122(3), 409-424.
- Ding, C., & Peng, H. (2003). Minimum redundancy feature selection from microarray gene expression data. *Proceedings of the 2003 IEEE Bioinformatics Conference (CSB 2003)*. (pp. 523-528).
- Dotterer, R. H. (1941). Ignorance and Equal Probability. *Philosophy of Science*, 8(3), 297-303.
- Dubois, D., & Prade, H. (1998). Soft computing, fuzzy logic, and artificial intelligence. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 2(1), 7-11.

- Dubois, D., & Prade, H. (1980). *Fuzzy sets and systems : theory and applications* (Mathematics in science and engineering, No. 144). New York : Academic Press.
- Dubois, D., & Prade, H. (1983). Unfair coins and necessity measures: Towards a possibilistic interpretation of histograms. *Fuzzy Sets and Systems*, 10(1), 15-20.
- Dubois, D., & Prade, H. (1988). *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. New York: Plenum Press.
- Dubois, D., & Prade, H. (1990). Rough Fuzzy Sets and Fuzzy Rough Sets. *International Journal of General Systems*, 17(2), 191-209.
- Dubois, D., & Prade, H. (1992). Putting Rough Sets and Fuzzy Sets Together. In R. Slowinski (Ed.), *Intelligent decision support : handbook of applications and advances of the rough sets theory* (pp. 203-232). Dordrecht, Boston: Kluwer Academic Publishers.
- Dubois, D., & Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, 90(2), 141-150.
- Dubois, D., & Prade, H. (2000). *Fundamentals of Fuzzy Sets* (The Handbooks of Fuzzy Sets Series, No. 7). Boston: Kluwer Academic Publishers.
- Duch, W. (2007). What Is Computational Intelligence and Where Is It Going? *Challenges for Computational Intelligence*, 1-13.
- Duch, W. (2005). Uncertainty of data, fuzzy membership functions, and multilayer perceptrons. *IEEE Transactions on Neural Networks*, 16(1), 10-23.
- Duch, W., Setiono, R., & Zurada, J. M. (2004). Computational intelligence methods for rule-based data understanding. *Proceedings of the IEEE*, 92(5), 771-805.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification* (2nd ed.). New York: John Wiley.
- Eisen, M. B., & Brown, P. O. (1999). DNA arrays for analysis of gene expression. In Sherman M. Weissman (Ed.), *Methods in Enzymology: cDNA Preparation and Characterization* (Volume 303 ed., pp. 179-205). Academic Press.
- Enders, W. (2004). *Applied econometric time series*. Hoboken, NJ: Wiley.
- Feuring, T., & Lippe, W.-M. (1999). The fuzzy neural network approximation lemma. *Fuzzy Sets and Systems*, 102(2), 227-236.
- Fischhoff, B., & Bruine de Bruin, W. (1999). Fifty-Fifty=50%. *Journal of Behavioral Decision Making*, 12 (2), 149-163.
- Flores-Sintas, A., Cadenas, J., & Martin, F. (1999). Membership functions in the fuzzy C-means algorithm. *Fuzzy Sets and Systems*, 101(1), 49-58.
- Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, 29(2- 3), 131-163.
- Gencay, R. (1998). The predictability of security returns with simple technical trading rules. *Journal of Empirical Finance*, 5(4), 347-359.
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P. et al. (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286(5439), 531-537.
- Gorman, R. P., & Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1(1), 75-89.
- Grossberg, S. (1972). Neural expectation: Cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes. *Kybernetik*, 10, 49-57.
- Grossberg, S. (1976a). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23, 121-134.
- Grossberg, S. (1976b). Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, 23, 187-202.

- Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, 87, 1-51.
- Guillaume, S. (2001). Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, 9(3), 426-443.
- Ha, T. M. (1997). The optimum class-selective rejection rule. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6), 608-615.
- Halgamuge, S. K., & Glesner, M. (1992). A Fuzzy-Neural Approach for Pattern Classification with the Generation of Rules based on Supervised Learning. In *Proceedings of the NEURO-NIMES 92: The 5th International Conference on Neural Networks and their Applications* (pp. 167-173). Nanterre Cedex: EC2.
- Halgamuge, S. K., & Glesner, M. (1994). Neural networks in designing fuzzy systems for real world applications. *Fuzzy Sets and Systems*, 65(1), 1-12.
- Hall, L. O. (1990). The choice of ply operator in fuzzy intelligent systems. *Fuzzy Sets and Systems*, 34(2), 135-144.
- Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann.
- Harvey, N. (1997). Confidence in judgment. *Trends in Cognitive Sciences*, 1(2), 78-82.
- Hayashi, I., Nomura, H., Yamasaki, H., & Wakami, N. (1992). Construction of fuzzy inference rules by NDF and NDFL. *International Journal of Approximate Reasoning*, 6(2), 241-266.
- Haykin, S. S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.
- Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.
- Hopfield, J. J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8, [Part 1: Biological Sciences]), 2554-2558.
- Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2), 251-257.
- Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.
- Höhle, U., & Rodabaugh, S. E. (1998). *Mathematics of Fuzzy Sets: Logic, Topology, and Measure theory* (The Handbooks of Fuzzy Sets Series, No. 3). Boston: Kluwer Academic Publishers.
- Inza, I., Sierra, B., Blanco, R., & Larrañaga, P. (2002). Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *Journal of Intelligent & Fuzzy Systems*, 12(1), 25.
- Ishibuchi, H., Tanaka, H., & Okada, H. (1994). Interpolation of fuzzy if-then rules by neural networks. *International Journal of Approximate Reasoning*, 10(1), 3-27.
- Jaeger, J., Sengupta, R., & Ruzzo, W. L. (2003). Improved Gene Selection for Classification of Microarrays. *Proceedings of the Pacific Symposium on Biocomputing* (Vol. 8, pp. 53-64).
- Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 4-37.
- Jang, J.-S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665-685.
- Jang, J.-S. R., & Sun, C.-T. (1995). Neuro-fuzzy modeling and control. *Proceedings of the IEEE*, 83(3), 378-406.
- Jang, J.-S. R., Sun, C.-T., & Mizutani Eiji. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River, NJ: Prentice Hall.
- Jensen, R., & Shen, Q. (2004). Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. *IEEE Transactions on Knowledge and Data Engineering*, 16(12), 1457-1471.
- Jensen, R., & Shen, Q. (2007a). Fuzzy-Rough Sets Assisted Attribute Selection. *IEEE Transactions on Fuzzy Systems*, 15(1), 73-89.

- Jensen, R., & Shen, Q. (2007b). New Approaches to Fuzzy-Rough Feature Selection. *IEEE Transactions on Fuzzy Systems*, in press.
- Jin, Y. (2000). Fuzzy modeling of high-dimensional systems: complexity reduction. *IEEE Transactions on Fuzzy Systems*, 8(2), 212-221.
- Johansen, T. A., & Babuška, R. (2003). Multiobjective identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 11(6), 847-860.
- Juang, C.-F., & Lin, C.-T. (1998). An online self-constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, 6(1), 12-32.
- Kandel, A. (1992). *Fuzzy expert systems*. Boca Raton, FL: CRC Press.
- Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (1995). *Essentials of neural science and behavior*. Norwalk, CT: Appleton & Lange.
- Kasabov, N. (1998). Evolving Fuzzy Neural Networks - Algorithms, Applications and Biological Motivation. In *Methodologies for the Conception, Design and Application of Soft Computing: Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems* (pp. 271-274). Singapore: World Scientific.
- Kasabov, N. (2001). Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 31(6), 902-918.
- Kasabov, N. K., & Song, Q. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2), 144-154.
- Kaynak, O., Jezernik, K., & Szeghegyi, A. (2002). Complexity reduction of rule based models: a survey. *FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on Fuzzy Systems* (Vol. 2, pp. 1216-1221).
- Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A Fuzzy K-Nearest Neighbor Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(4), 580-585.
- Keller, J. M., & Tahani, H. (1992). Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks. *International Journal of Approximate Reasoning*, 6(2), 221-240.
- Keller, J. M., Yager, R. R., & Tahani, H. (1992). Neural network implementation of fuzzy logic. *Fuzzy Sets and Systems*, 45(1), 1-12.
- Kim, T.-Y., & Ko, H. (2005). Bayesian fusion of confidence measures for speech recognition. *IEEE Signal Processing Letters*, 12 (12), 871-874.
- Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. *IJCNN International Joint Conference on Neural Networks* (pp. 1-6 vol.1).
- Klir, G. J. (1990). A Principle of uncertainty and information invariance. *International Journal of General Systems*, 17(2-3), 249-275.
- Klir, G. J. (1995). Principles of uncertainty: What are they? Why do we need them? *Fuzzy Sets and Systems*, 74(1), 15-31.
- Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ: Prentice Hall.
- Knuth, D. E. (1997). *Art of Computer Programming, Volume 2: Seminumerical Algorithms* (3rd ed.). Canada: Addison-Wesley.
- Koczy, L., & Hirota, K. (1993). Ordering, distance and closeness of fuzzy sets. *Fuzzy Sets and Systems*, 59(3), 281-293.
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273-324.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.
- Kohonen, T. (1989). *Self-Organization and Associative Memory* (3rd ed.). Berlin, New York: Springer-Verlag.

- Kosko, B. (1992). *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence*. Englewood Cliffs, NJ: Prentice Hall.
- Kreinovich, V., Nguyen, H. T., & Yeung, Y. (2000). Fuzzy systems are universal approximators for a smooth function and its derivatives. *International Journal of Intelligent Systems*, 15(6), 565-574.
- Kryszkiewicz, M. (1998). Rough set approach to incomplete information systems. *Information Sciences*, 112(1-4), 39-49.
- Kugiumtzis, D., Lillekjendlie, B., & Christophersen, N. (1994). Chaotic time series part I: Estimation of some invariant properties in state space. *Modeling, Identification and Control*, 15(4), 205-224.
- Kwak, N., & Chong-Ho Choi. (2002). Input feature selection by mutual information based on Parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12), 1667-1671.
- Lafarga, M., Berciano, M. T., Saurez, I., Andres, M. A., & Berciano, J. (1993). Reactive astroglia-neuron relationships in the human cerebellar cortex: A quantitative, morphological and immunocytochemical study in Creutzfeldt-Jakob disease. *International Journal of Developmental Neuroscience*, 11(2), 199-213.
- Lee, C. C. (1990a). Fuzzy logic in control systems: fuzzy logic controller. I. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2), 404-418.
- Lee, C. C. (1990b). Fuzzy logic in control systems: fuzzy logic controller. II. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2), 419-435.
- Lee, E. S., & Zhu, Q. (1995). *Fuzzy and Evidence Reasoning* (Studies in Fuzziness, No. 6). Heidelberg: Physica-Verlag.
- Lee, M., Lee, S.-Y., & Park, C. H. (1994). A new neuro-fuzzy identification model of nonlinear dynamic systems. *International Journal of Approximate Reasoning*, 10(1), 29-44.
- Li, C., & Cheng, K.-H. (2007). Recurrent neuro-fuzzy hybrid-learning approach to accurate system modeling. *Fuzzy Sets and Systems*, 158(2), 194-212.
- Li, M., & Sethi, I. K. (2006a). Confidence-Based Active Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8), 1251-1261.
- Li, M., & Sethi, I. K. (2006b). Confidence-based classifier design. *Pattern Recognition*, 39(7), 1230-1240.
- Li, R.-P., Mukaidono, M., & Turksen, I. B. (2002). A fuzzy neural network for pattern classification and feature selection. *Fuzzy Sets and Systems*, 130(1), 101-108.
- Li, Y., & Guan, C. (2006). An Extended EM Algorithm for Joint Feature Extraction and Classification in Brain-Computer Interfaces. *Neural Computation*, 18(11), 2730-2761.
- Lin, C.-J., & Lin, C.-T. (1997). An ART-based fuzzy adaptive learning control network. *IEEE Transactions on Fuzzy Systems*, 5(4), 477-496.
- Lin, C.-T. (1995). A neural fuzzy control system with structure and parameter learning. *Fuzzy Sets and Systems*, 70(2-3), 183-212.
- Lin, C.-T., & Lee, C. S. G. (1991). Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, 40(12), 1320-1336.
- Lin, C.-T., & Lee, C. S. G. (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River, NJ: Prentice Hall.
- Lin, T. Y., & Cercone, N. (1997). *Rough Sets and Data Mining: Analysis of Imprecise Data*. Boston, London, Dordrecht: Kluwer Academic Publishers.
- Lin, Z., & Brannigan, A. (2003). Advances in the Analysis of Non-stationary Time Series: An Illustration of Cointegration and Error Correction Methods in Research on Crime and Immigration. *Quality and Quantity*, 37(2), 151-168.
- Lindskog, P. (1997). Fuzzy identification from a grey box modeling point of view. In H. Hellendoorn, & D. Driankov (Eds.), *Fuzzy Model Identification: Selected Approaches*. Heidelberg: Springer-Verlag.

- Lingras, P. (1998). Comparison of neofuzzy and rough neural networks. *Information Sciences*, 110(3-4), 207-215.
- Liu, F., Ng, G. S., Quek, C., & Loh, T. F. (2006). Artificial Ventilation Modeling using Neuro-Fuzzy Hybrid System. *International Joint Conference on Neural Networks (IJCNN '06)* (pp. 2859-2864).
- Lotfi, A., Andersen, H. C., & Tsoi, A. C. (1996). Matrix formulation of fuzzy rule-based systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 26(2), 332-340.
- Lozowski, A., Cholewo, T. J., & Zurada, J. M. (1996). Crisp rule extraction from perceptron network classifiers. In *Proceedings of International Conference on Neural Networks*, vol. Plenary, Panel and Special Sessions (pp. 94-99). Washington, D.C.
- Maass, W., & Bishop, C. M. (1999). *Pulsed Neural Networks*. Cambridge, Mass.: MIT Press.
- Maguire, L. P., Roche, B., McGinnity, T. M., & McDaid, L. J. (1998). Predicting a chaotic time series using a fuzzy neural network. *Information Sciences*, 112(1-4), 125-136.
- Mamdani, E. H. (1976). Advances in the linguistic synthesis of fuzzy controllers. *International Journal of Man-Machine Studies*, 8(6), 669-678.
- Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1), 1-13.
- Masson, M.-H., & Denoeux, T. (2006). Inferring a possibility distribution from empirical data. *Fuzzy Sets and Systems*, 157(3), 319-340.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.
- Medasani, S., Kim, J., & Krishnapuram, R. (1998). An overview of membership function generation techniques for pattern recognition. *International Journal of Approximate Reasoning*, 19(3-4), 391-417.
- Mendel, J. M. (2001). *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice Hall.
- Mikut, R., Jakel, J., & Groll, L. (2005). Interpretability issues in data-based learning of fuzzy systems. *Fuzzy Sets and Systems*, 150(2), 179-197.
- Miller, I., & Miller, M. (2004). *John E. Freund's Mathematical Statistics with Applications* (7th ed.). Upper Saddle River, NJ: Pearson Prentice Hall.
- Mitra, S., & Hayashi, Y. (2000). Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11(3), 748-768.
- Mizumoto, M., & Zimmermann, H.-J. (1982). Comparison of fuzzy reasoning methods. *Fuzzy Sets and Systems*, 8(3), 253-283.
- Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting*, 17(5-6), 441-470.
- Moser, B. (1999). Sugeno controllers with a bounded number of rules are nowhere dense. *Fuzzy Sets and Systems*, 104(2), 269-277.
- Müller, B., Reinhardt, J., & Strickland, M. T. (1995). *Neural Networks: An Introduction* (2nd ed.). New York: Springer-Verlag.
- Nakanishi, H., Turksen, I. B., & Sugeno, M. (1993). A review and comparison of six reasoning methods. *Fuzzy Sets and Systems*, 57(3), 257-294.
- Nauck, D. D. (2003). Measuring interpretability in rule-based classification systems. *FUZZ '03. The 12th IEEE International Conference on Fuzzy Systems* (Vol. 1, pp. 196-201).
- Nauck, D., Klawonn, F., & Kruse, R. (1997). *Foundations of Neuro-Fuzzy Systems*. New York: John Wiley & Sons, Inc.
- Nedjah, N., & Mourelle, L. M. (2005). *Fuzzy systems engineering : theory and practice* (Studies in fuzziness and soft computing, No. 181). Berlin, New York: Springer.

- Newman, D. J., Hettich, S., Blake, C. L., & Merz, C. J. (1998). *UCI Repository of machine learning databases*. University of California, Irvine, Dept. of Information and Computer Sciences: Web site: <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- Nguyen, H. T., & Sugeno, M. (1998). *Fuzzy Systems: Modeling and Control* (The Handbooks of Fuzzy Sets Series, No. 2). Boston: Kluwer Academic Publishers.
- Nowicki, R., & Rutkowski, L. (2002). Rough-neuro-fuzzy system for classification. *Proceedings of the 2002 International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '02)* (Vol. 2, pp. 463-466).
- Packard, N. H., Crutchfield, J. P., Farmer, J. D., & Shaw, R. S. (1980). Geometry from a time series. *Physical Review Letters*, 45(9), 712-715.
- Pal, S. K., & Skowron, A. (1999). *Rough fuzzy hybridization : a new trend in decision-making*. Singapore, New York: Springer.
- Panella, M., & Gallo, A. S. (2005). An input-output clustering approach to the synthesis of ANFIS networks. *IEEE Transactions on Fuzzy Systems*, 13(1), 69-81.
- Pantazopoulos, K. N., Tsoukalas, L. H., Bourbakis, N. G., Brun, M. J., & Houstis, E. N. (1998). Financial prediction and trading strategies using neurofuzzy approaches. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 28(4), 520-531.
- Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3), 1065-1076.
- Patrick, E. A., & Fischer III, F. P. (1970). A generalized k-nearest neighbor rule. *Information and Control*, 16(2), 128-152.
- Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences*, 11(5), 341-356.
- Pawlak, Z. (1984). Rough classification. *International Journal of Human-Computer Studies*, 20(5), 469-483.
- Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Dordrecht, Boston: Kluwer Academic Publishers.
- Pawlak, Z. (2002). Rough sets and intelligent data analysis. *Information Sciences*, 147(1-4), 1-12.
- Pawlak, Z., & Skowron, A. (2007a). Rough sets and Boolean reasoning. *Information Sciences*, 177(1), 41-73.
- Pawlak, Z., & Skowron, A. (2007b). Rough sets: Some extensions. *Information Sciences*, 177(1), 28-40.
- Pawlak, Z., & Skowron, A. (2007c). Rudiments of rough sets. *Information Sciences*, 177(1), 3-27.
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 1226-1238.
- Plummer, T., & Ridley, A. (2003). *Forecasting Financial Markets: The Psychology of Successful Investing* (4th ed.). London: Kogan Page.
- Poole, D. L., Mackworth, A. K., & Goebel, R. (1998). *Computational Intelligence: A Logical Approach*. New York: Oxford University Press.
- Quah, K. H., & Quek, C. (2007). MCES: A Novel Monte Carlo Evaluative Selection Approach for Objective Feature Selections. *IEEE Transactions on Neural Networks*, 18(2), 431-448.
- Quek, C., & Zhou, R. W. (1999). POPFNN-AAR(S): a pseudo outer-product based fuzzy neural network. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 29(6), 859-870.
- Quek, C., & Zhou, R. W. (2001). The POP learning algorithms: reducing work in identifying fuzzy rules. *Neural Networks*, 14(10), 1431-1445.
- Quilan, J. R. (1983). Learning efficient classification procedures and their applications to chess end games. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (Vol. 1, pp. 463-482). Palo Alto, CA: Tioga Pub. Co.

- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106.
- Richard, M. D., & Lippmann, R. P. (1991). Neural Network Classifiers Estimate Bayesian a posteriori Probabilities. *Neural Computation*, 3(4), 461-483.
- Roadknight, C. M., Balls, G. R., Mills, G. E., & Palmer-Brown, D. (1997). Modeling complex environmental data. *IEEE Transactions on Neural Networks*, 8(4), 852-862.
- Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1, pp. 318-362). Cambridge, Massachusetts: MIT Press.
- Russell, S. J., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.
- Rutkowski, L., & Cpałka, K. (2003). Flexible neuro-fuzzy systems. *IEEE Transactions on Neural Networks*, 14(3), 554-574.
- Saad, E. W., Prokhorov, D. V., & Wunsch II, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks*, 9(6), 1456-1470.
- Schummer, M., Ng, W. V., Bumgarner, R. E., Nelson, P. S., Schummer, B., Bednarski, D. W. et al. (1999). Comparative hybridization of an array of 21 500 ovarian cDNAs for the discovery of genes overexpressed in ovarian carcinomas. *Gene*, 238(2), 375-385.
- Schölkopf, B., Burges, C. J. C., & Smola, A. J. (1999). *Advances in kernel methods : support vector learning*. Cambridge, Mass: MIT Press.
- Shang, C., & Shen, Q. (2002). Rough Feature Selection for Neural Network Based Image Classification. *International Journal of Image & Graphics: International Journal of Image & Graphics*, 2(4), 541.
- Shann, J. J., & Fu, H. C. (1995). A fuzzy neural network for rule acquiring on fuzzy control systems. *Fuzzy Sets and Systems*, 71(3), 345-357.
- Shen, Q., & Chouchoulas, A. (2002). A rough-fuzzy approach for generating classification rules. *Pattern Recognition*, 35(11), 2425-2438.
- Sima, J., & Orponen, P. (2003). General-Purpose Computation with Neural Networks: A Survey of Complexity Theoretic Results. *Neural Computation*, 15(12), 2727-2778.
- Sjoberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y. et al. (1995). Nonlinear black-box modeling in system identification: a unified overview: Trends in System Identification. *Automatica*, 31(12), 1691-1724.
- Skowron, A., & Rauszer Cecylia. (1992). The discernibility matrices and functions in information systems. In R. Slowinski (Ed.), *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory* (pp. 331-362). Dordrecht, Boston, London: Kluwer Academic Publishers.
- Slowinski, R. (1998). *Fuzzy Sets in Decision Analysis, Operations Research and Statistics* (The handbooks of fuzzy sets series, No. 1). Boston: Kluwer Academic Publishers.
- Song, Q., & Kasabov, N. K. (2005). NFI: a neuro-fuzzy inference method for transductive reasoning. *IEEE Transactions on Fuzzy Systems*, 13(6), 799-808.
- Sugeno, M., & Kang, G. T. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1), 15-33.
- Sugeno, M., & Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1), 7.
- Swiniarski, R. W., & Skowron, A. (2003). Rough set methods in feature selection and recognition. *Pattern Recognition Letters*, 24(6), 833-849.

- Söderström, T., & Stoica, P. (1989). *System identification*. New York: Prentice Hall.
- Takagi, H. (2000). Fusion Technology of Neural Networks and Fuzzy Systems: A Chronicled Progression from the Laboratory to Our Daily Lives. *International Journal of Applied Mathematics and Computer Science*, 10(4), 647-673.
- Takagi, H., & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning*, 5(3), 191-212.
- Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1), 116-132.
- Takens, F. (1981). Detecting strange attractors in turbulence. In D. A. Rand, & L.-S. Young (Eds), *Dynamical Systems and Turbulence* (pp. 366-381). New York: Springer-Verlag.
- Tan, G. K. (1997). *Feasibility of predicting congestion states with neural network models*. Final Year Project Report, School of Civil and Structural Engineering, Nanyang Technological University, Singapore.
- Tan, T. Z., Quek, C., & Ng, G. S. (2005). Ovarian cancer diagnosis by hippocampus and neocortex-inspired learning memory structures. *Neural Networks*, 18(5-6), 818-825.
- Tay, T.-T., & Tan, S. W. (1997). Fuzzy system as parameter estimator of nonlinear dynamic functions. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 27(2), 313-326.
- Tikk, D., & Baranyi, P. (2003). Exact trade-off between approximation accuracy and interpretability: solving the saturation problem for certain FRBSs. In J. Casillas, O. Cordón, F. Herrera, & L. Magdalena (Eds.), *Interpretability Issues in Fuzzy Modeling* (pp. 587-604). Berlin: Springer-Verlag.
- Tikk, D., Kóczy, L. T., & Gedeon, T. D. (2003). A survey on universal approximation and its limits in soft computing techniques. *International Journal of Approximate Reasoning*, 33(2), 185-202.
- Tsukamoto, Y. (1979). An approach to fuzzy reasoning method. In M. M. Gupta, R. K. Ragade, & R. R. Yager (Eds.), *Advances in fuzzy set theory and applications* (pp. 137-149). Amsterdam: North-Holland Pub. Co.
- Tung, W. L., & Quek, C. (2002a). DIC: A Novel Discrete Incremental Clustering Technique for the Derivation of Fuzzy Membership Functions. In *Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence* (pp. 178-187). Springer-Verlag.
- Tung, W. L., & Quek, C. (2002b). GenSoFNN: a generic self-organizing fuzzy neural network. *IEEE Transactions on Neural Networks*, 13(5), 1075-1086.
- Tung, W. L., & Quek, C. (2005). GenSo-FDSS: a neural-fuzzy decision support system for pediatric ALL cancer subtype identification using gene expression data. *Artificial Intelligence in Medicine*, 33(1), 61-88.
- Turksen, I. B., & Zhong, Z. (1990). An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets. *Fuzzy Sets and Systems*, 34(3), 323-346.
- Vapnik, V. N. (1998). *Statistical learning theory* (Adaptive and learning systems for signal processing, communications, and control). New York: Wiley.
- Wang, L.-X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22(6), 1414-1427.
- Wang, L.-X. (1998). Universal approximation by hierarchical fuzzy systems. *Fuzzy Sets and Systems*, 93(2), 223-230.
- Wang, L.-X., & Wei, C. (2000). Approximation accuracy of some neuro-fuzzy approaches. *IEEE Transactions on Fuzzy Systems*, 8(4), 470-478.
- Wang, W., & Zhang, Y. (2007). On fuzzy cluster validity indices: Theme: Data Analysis. *Fuzzy Sets and Systems*, 158(19), 2095-2117.
- Wang, Y., Makedon, F. S., Ford, J. C., & Pearlman, J. (2005). HykGene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. *Bioinformatics*, 21(8), 1530-1537.

- Whitney, A. W. (1971). A Direct Method of Nonparametric Measurement Selection. *IEEE Transactions on Computers*, 20(9), 1100-1103.
- Widrow, B., & Lehr, M. A. (1990). 30 years of adaptive neural networks: perceptron, Madaline, and backpropagation. *Proceedings of the IEEE*, 78(9), 1415-1442.
- Widrow, B., & Hoff Jr., M. E. (1960). Adaptive switching circuits. *IRE Western Electric Show and Convention Record*, 4, 96-104.
- Yager, R. R., Ovchinnikov, S., Tong, R. M., & Nguyen, H. T. (1987). *Fuzzy sets and applications: Selected Papers by L. A. Zadeh*. New York: John Wiley.
- Yager, R. R. (1994). Modeling and formulating fuzzy knowledge bases using neural networks. *Neural Networks*, 7(8), 1273-1283.
- Yao, Y. Y. (1996). Two views of the theory of rough sets in finite universes. *International Journal of Approximate Reasoning*, 15(4), 291-317.
- Yao, Y. Y. (1998). A comparative study of fuzzy sets and rough sets. *Information Sciences*, 109(1-4), 227-242.
- Yen, J., Wang, L., & Gillespie, C. W. (1998). Improving the interpretability of TSK fuzzy models by combining global learning and local learning. *IEEE Transactions on Fuzzy Systems*, 6(4), 530-537.
- Yeung, D. S., & Tsang, E. C. C. (1997). A comparative study on similarity-based fuzzy reasoning methods. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 27(2), 216-227.
- Ying, H. (1994). Sufficient conditions on general fuzzy systems as function approximators. *Automatica*, 30(3), 521-525.
- Ying, H., & Chen, G. (1997). Necessary Conditions for Some Typical Fuzzy Systems as Universal Approximators. *Automatica*, 33(7), 1333-1338.
- Ying, H., Ding, Y., Li, S., & Shao, S. (1999). Comparison of necessary conditions for typical Takagi-Sugeno and Mamdani fuzzy systems as universal approximators. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 29(5), 508-514.
- Yoma, N. B., Carrasco, J., & Molina, C. (2005). Bayes-based confidence measure in speech recognition. *IEEE Signal Processing Letters*, 12(11), 745-748.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-353.
- Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, 3(1), 28-44.
- Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-I,II,III. *Information Sciences*, 8(3), 199-249; 8(4), 301-357; 9(1), 43-80.
- Zadeh, L. A. (1976). A fuzzy-algorithmic approach to the definition of complex or imprecise concepts. *International Journal of Man-Machine Studies*, 8(3), 249-291.
- Zadeh, L. A. (1978a). PRUF - A Meaning Representation Language for Natural Language. *International Journal of Man-Machine Studies*, 10, 395-460.
- Zadeh, L. A. (1978b). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1), 3-28.
- Zadeh, L. A. (1979). A Theory of Approximate Reasoning. In J. Hayes, D. Michie, & L. I. Mikulich (Eds.), *Machine Intelligence* (Vol. 9, pp. 149-194). New York: Halstead Press.
- Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3),
- Zadeh, L. A. (1996). Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2), 103-111.
- Zadeh, L. A. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90(2), 111-127.
- Zadeh, L. A. (2006). Generalized theory of uncertainty (GTU)--principal concepts and ideas: The Fuzzy Approach to Statistical Analysis. *Computational Statistics & Data Analysis*, 51(1), 15-46.

- Zeng, X.-J., & Singh, M. G. (1996). Approximation accuracy analysis of fuzzy systems as function approximators. *IEEE Transactions on Fuzzy Systems*, 4(1), 44-63.
- Zhou, R. W., & Quek, C. (1996). POPFNN: A Pseudo Outer-product Based Fuzzy Neural Network. *Neural Networks*, 9(9), 1569-1581.
- Zhou, X., & Mao, K. (2005). Gene Selection of DNA Microarray Data Based on Regularization Networks. In M. Gallagher, J. Hogan, & F. Maire (Eds.), *Intelligent Data Engineering and Automated Learning - IDEAL 2005, Proceedings Lecture Notes in Computer Science* (Vol. 3578/2005, pp. 414-421).
- Ziarko, W. (1993). Variable precision rough set model. *Journal of Computer and System Sciences*, 46(1), 39-59.
- Zimmermann, H.-J. (1999). *Practical Applications of Fuzzy Technologies* (The Handbooks of Fuzzy Sets Series, No. 6). Boston: Kluwer Academic Publishers.
- Zurada, J. M. (1992). *Introduction to Artificial Neural Systems*. St. Paul: West.

Author's Publications

International Journals

- *Ang, K. K., & Quek, C. (2000). Improved MCMAC with momentum, neighborhood, and averaged trapezoidal output. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 30(3), 491-500.
- *Ang, K. K., Quek, C., & Wahab, A. (2002). MCMAC-CVT: a novel on-line associative memory based CVT transmission control system. *Neural Networks*, 15(2), 219-236.
- *Ang, K. K., Quek, C., & Pasquier, M. (2003). POPFNN-CRI(S): pseudo outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 33(6), 838-849.
- Ang, K. K., & Quek, C. (2005). RSPOP: Rough Set-Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. *Neural Computation*, 17(1), 205-243.
- Ang, K. K., & Quek, C. (2006). Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach. *IEEE Transactions on Neural Networks*, 17(5), 1301-1315.

International Conferences

- Ang, K. K., & Quek, C. (2005). Stock Trading using PSEC and RSPOP: A novel evolving rough set-based neuro-fuzzy approach. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC '05)* (Vol. 2, pp. 1032-1039).
- Ang, K. K., & Quek, C. (2006). Rough Set-based Neuro-Fuzzy System. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '06)* (pp. 742-749).

Journal Manuscripts under preparation/review

- Ang, K. K., & Quek, C. (2008). SPSEC: Supervised Pseudo-Self Evolving Cerebellar fuzzy membership generation algorithm.
- Ang, K. K., & Quek, C. (2008). MIRS: Mutual Information-based Rough Set Reduction feature selection algorithm using wrapper approach.
- Ang, K. K., & Quek, C. (2008). Confidence-based classification using Rough set-based Neuro-Fuzzy System.

Book Chapter in press

- Ang, K. K., & Quek, C. (2008). Rough set-based Neuro-Fuzzy System. In J. R. Rabuñal, J. Dorado, & A. Pazos (Eds), *Encyclopedia of Artificial Intelligence*. Information Science Reference.

* Papers published on research that is apart from this Ph.D. project.