University of Technology, Sydney

Fuzzy Transfer Learning for Financial Early Warning System

A Thesis Submitted for the Degree of

Doctor of Philosophy

By

Vahid Behbood

November, 2012

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Production Note:
Signature removed prior to publication.

DEDICATION

To my beloved Mother, for her prayers for me.

To the soul of my Father, the first to teach me.

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my PhD principal supervisor, Prof. Jie Lu, for offering me an opportunity to begin my research study three years ago. Thank you for your continuous support, encouragement, precious guidance and patience throughout my study. Thank you for your accurate critical comments and suggestions, which have strengthened this study significantly. Your strict academic attitude, generous personality and conscientious working style have influenced me deeply, and will be of great benefit to me in my future research work and life. I have learned a lot more from you than you realize. I also would like to address my sincere thanks to my co-supervisor, A/Prof. Guangquan Zhang for his knowledgeable suggestions and valuable advice, which have greatly improved the final quality of my PhD thesis.

Most of all, I would like to express my deepest appreciation to my Mother. Thank you for supporting me and always being there for me during my ups and downs. Pursuing a PhD was always a long term challenge for me and a dream of my mother's. This dream would have not been achieved without the love, great empathy and kind assistance of my mother. Very special thanks to my mother who believes in me more than I do in myself.

I would like to express my highest appreciation to my sister, Leila, for her unconditional encouragement and support throughout my study. It would have been impossible for me to complete my study without her help. She shared all my pain, sorrow and depression in the most difficult time of my research. Her optimistic view of life encourages me a lot, not only in my research but also in my life.

I would also like to thank Ms. Sue Felix for helping me to identify and correct grammar, syntax and presentation problems in my thesis. I am grateful to the School of Software in the Faculty of Engineering and Information Technology at the University of Technology, Sydney. This study was fully supported by the International Postgraduate Research Scholarship (IPRS).

Finally, I wish to express my appreciation to all my friends and the members (especially Mr. Qusai Shambour) of the Decision Systems & e-Service Intelligence (DeSI) Lab in the Centre for Quantum Computation and Intelligent Systems (QCIS) for their help, participation and valuable comments in every presentation I made during my study.

TABLE OF CONTENTS

CERTIFICATE OF AUTHORSHIP/ORIGINALITY ..... *I*

DEDICATION ..... *II*

ACKNOWLEDGEMENTS ..... *III*

LIST OF FIGURES ..... *XI*

LIST OF TABLES ..... *XIV*

ABSTRACT ..... *XVIII*

CHAPTER 1: Introduction ***1***

1.1 Background **1**

1.2 Research Challenges **6**

1.3 Research Objectives **9**

1.4 Research Contributions **12**

1.5 Research Methodology and Process **14**

1.5.1 Research Methodology 15

1.5.1.1 Awareness of Problem 15

1.5.1.2 Suggestion 15

1.5.1.3 Development 16

1.5.1.4 Evaluation 16

1.5.1.5 Conclusion 16

1.5.2 Research Process 17

1.6 Thesis Structure **17**

1.7 Publications Related to This Thesis **20**

CHAPTER 2: Literature Review ***22***

2.1 Bank Failure Prediction and Early Warning System 22

2.2 Feature Selection 25

2.3 Financial Failure Prediction: Classical Statistical Methods 28

2.4 Financial Failure Prediction: Intelligent Methods 31

2.4.1 Case-based Reasoning 32

2.4.2 Decision Tree 33

2.4.3 Support Vector Machines 33

2.4.4 Fuzzy Rule Based Classifier 34

2.4.5 Neural Network 35

2.4.6 Ensemble-based and Hybrid Methods 36

2.4.7 Fuzzy Neural Network 39

2.5 Class Imbalance Problem and Solutions 44

2.5.1 Data Level Methods 45

2.5.2 Algorithm Level Methods 47

2.5.3 Cost-sensitive Learning Methods 48

2.5.4 Boosting Approaches 48

2.5.5 Evaluation Measures 49

2.6 Transfer Learning 52

2.6.1 Definitions and Notations 55

2.6.2 Transductive Transfer Learning 57

2.6.2.1 Instance Weighting for Covariate Shift Methods 59

2.6.2.2 Self-labeling Methods 60

2.6.2.3 Feature Representation Methods 62

2.6.2.4 Cluster-based Learning Methods 65

CHAPTER 3: Adaptive Inference-based Fuzzy Neural Network 67

3.1 Introduction 67

3.2 Concepts and Definitions 68

3.2.1 The Problem of Imbalanced Data Set 68

3.2.2 The Synthetic Minority Oversampling Technique 69
3.2.3 Accuracy Evaluation 69
3.2.4 Discrete Incremental Clustering 70

3.3 Fuzzy Neural Network Structure and Rule Generation Algorithm 71
3.3.1 Fuzzy Neural Network Prediction Approach Outline 71
3.3.2 Structure of Fuzzy Neural Network 72
3.3.3 Rule Generation Algorithm 72

3.4 Adaptive Inference-based Learning Algorithm 75

3.5 Empirical Results Analysis 79
3.5.1 Data Sets 80
3.5.2 Research Design and Pre-processing 80
3.5.3 Experiment using Different Scenarios 82
3.5.4 Benchmark Against Other Models 89
3.5.4.1 Benchmark using Pre-processed Data 91
3.5.4.2 Benchmark using Imbalanced Data 95
3.5.4.3 GM_Error Function Analysis 96
3.5.5 Fuzzy Rule Analysis 100

3.6 Summary 101

CHAPTER 4: Multi Step Fuzzy Bridge Refinement Domain Adaptation 104

4.1 Introduction 104

4.2 Concepts and Definitions 107

4.3 Fuzzy Bridged Refinement Domain Adaptation 109
4.3.1 Bridged Refinement-based Theory 109
4.3.2 Multi-Step Fuzzy Bridged Refinement-based Algorithm 118

4.4 Experiments and Empirical Analysis 125
4.4.1 Data Sets 126
4.4.2 Research Design and Comparison 127
4.4.3 Experiment Results Analysis 129

4.4.3.1 Results Analysis using Different Settings 129
4.4.3.2 Results Analysis Comparing MSFBR and MSBR 131
4.4.3.3 Results Analysis using Different Prediction Methods 132
4.4.4 Parameter Sensitivity 133

4.5 Summary 135

CHAPTER 5: Fuzzy Feature Alignment-based Cross-Domain Adaptation 137

5.1 Introduction 137

5.2 Problem Setting and Definitions 140

5.3 The Fuzzy Cross-Domain Adaptation Approach 148
5.3.1 Phase One 148
5.3.2 Phase Two 149
5.3.3 Phase Three 150
5.3.3.1 Fuzzy Spectral Feature Alignment 150
5.3.3.2 Fuzzy Spectral Feature Graph Structure 151
5.3.3.3 Fuzzy Correlation Coefficient 152
5.3.3.4 Fuzzy Spectral Feature Alignment Algorithm 153
5.3.4 Phase Four 156
5.3.4.1 Feature Selection 156
5.3.4.2 Fuzzy Genetic Feature Weighting Algorithm 157
5.3.5 Phase Five 158

5.4 Empirical Analysis for Bank Failure Prediction 159
5.4.1 Data Sets 159
5.4.1.1 Synthetic Data Set 159
5.4.1.2 Real Data Set 160
5.4.2 Research Design 161
5.4.2.1 Experiment Design using Synthetic Data Set 162
5.4.2.2 Experiment Design using Real World Financial Data Sets 163
5.4.3 Experiment Results Analysis 165
5.4.3.1 Experiment Results Analysis using Synthetic Data Sets 165

5.4.3.2 Experiment Results Analysis using Real World Financial Data Sets 168
5.5 Summary ... **176**
CHAPTER 6: Case Study: Australian Banks Experience ... ***178***
6.1 Introduction ... **178**
6.2 Problem Setting and Definitions ... **179**
6.3 Modelling ... **181**
6.3.1 Fuzzy Bridged Refinement Domain Adaptation (First Approach) ... 181
6.3.1.1 Multi-Step Fuzzy Bridge Refinement Algorithm Type I ... 186
6.3.1.2 Multi-Step Fuzzy Bridge Refinement Algorithm Type II ... 189
6.3.1.3 Multi-Step Fuzzy Bridge Refinement Algorithm Type III ... 191
6.3.2 Feature Alignment-based Cross Domain Adaptation (Second Approach) 195
6.3.2.1 Phase One ... 196
6.3.2.2 Phase Two ... 196
6.3.2.3 Phase Three ... 197
6.3.2.4 Phase Four ... 199
6.3.2.5 Phase Five ... 201
6.4 Experiments and analysis ... **201**
6.4.1 Data Sets ... 202
6.4.2 Research Design and Comparison ... 203
6.4.2.1 Experiment Design for Bridged Refinement-based Domain Adaptation (First Approach) ... 204
6.4.2.2 Experiment Design for Feature Alignment-based Cross Domain Adaptation (Second Approach) ... 206
6.4.3 Experiment Results Analysis for Fuzzy Bridged Refinement Domain Adaptation (First approach) ... 207
6.4.4 Experiment Results Analysis for Feature Alignment-based Cross Domain Adaptation (Second Approach) ... 213
6.5 Summary ... **217**
CHAPTER 7: Conclusions and Future Study ... ***220***
7.1 Conclusions ... **220**

7.2 Future Study..... 222

References 227

Appendix: Abbreviations..... 244

LIST OF FIGURES

FIGURE 1.1: THE GENERAL METHODOLOGY OF DESIGN RESEARCH (NIU ET AL. 2009) ... *15*
FIGURE 1.2: THESIS STRUCTURE ... *19*
FIGURE 2.1: NUMBER OF FAILED BANKS IN UNITED STATES ... *24*
FIGURE 2.2: TOTAL ESTIMATED LOST (THOUSANDS DOLLAR) ... *24*
FIGURE 2.3: ROC CURVES FOR TWO DIFFERENT MODELS ... *52*
FIGURE 2.4: DIFFERENT LEARNING PROCESSES BETWEEN TRADITIONAL MACHINE LEARNING AND TRANSFER LEARNING ... *54*
FIGURE 2.5: DIFFERENT LEARNING PROCESSES BETWEEN TRADITIONAL MACHINE LEARNING AND TRANSFER LEARNING ... *55*
FIGURE 3.1: OUTLINE OF THE PROPOSED FUZZY NEURAL NETWORK PREDICTION APPROACH ... *71*
FIGURE 3.2: STRUCTURE OF PROPOSED FUZZY NEURAL NETWORK ... *73*
FIGURE 3.3: DATA SET 1, SMOTE, SCENARIO 1 ... *83*
FIGURE 3.4: DATA SET 1, SMOTE, SCENARIO 6 ... *83*
FIGURE 3.5: DATA SET 2, SMOTE, SCENARIO 1 ... *83*
FIGURE 3.6: DATA SET 2, SMOTE, SCENARIO 6 ... *84*
FIGURE 3.7: AVERAGE ACCURACY OF ALL SCENARIOS ON BOTH DATA SETS ... *86*
FIGURE 3.8: ACCURACY FOR TWO DATA SETS USING SMOTE IN ALL SCENARIOS ... *87*
FIGURE 3.9: ACCURACY FOR TWO DATA SETS USING DOWN-SAMPLING IN ALL SCENARIOS ... *88*
FIGURE 3.10 : STANDARD DEVIATION FOR TWO DATA SETS USING SMOTE IN ALL SCENARIOS ... *88*
FIGURE 3.11: STANDARD DEVIATION FOR TWO DATA SETS USING DOWN-SAMPLING IN ALL SCENARIOS ... *89*
FIGURE 3.12: THE TREND OF DIFFERENCE IN THE PROPOSED APPROACH ACCURACY AGAINST OTHER FNNS ACCURACY ... *94*

Figure 3.13: The Trend of RMSE and GM_error by Increasing The FN 97
Figure 3.14: The First Differential of GM_error and RMSE by Increasing The FN 98
Figure 3.15: Fuzzy Sets Calculated by DIC on CV1 Group in Scenario 1 103
Figure 4.1: Multi-Step Fuzzy Bridged Refinement-based Domain Adaptation 123
Figure 4.2: Accuracy of Proposed Algorithm using 16 Experiments for 5, 8 and 10 Years Ahead Prediction 131
Figure 4.3: The Accuracy of Four Settings of FNN_ MSFRB using Different Value of K 134
Figure 4.4: The Accuracy of Four Settings of FNN_ MSFRB using Different Steps 135
Figure 4.5: The Accuracy of FNN_MSFBR using Different Value of α 135
Figure 5.1: Diagrams of Domains 141
Figure 5.2: Membership Function of Feature One of Source domain 143
Figure 5.3: Membership Function of Feature Two of Source domain 144
Figure 5.4: Membership Function of Feature Three of Source domain 144
Figure 5.5: Membership Function of Feature One of Target Domain 145
Figure 5.6: Membership Function of Feature Two of Target Domain 146
Figure 5.7 : Membership Function of Feature Three of Target Domain 146
Figure 5.8: Membership Function of Feature Four of Target Domain 147
Figure 5.9: Membership Function of Feature Five of Target Domain 148
Figure 5.10: FNN Structure of Example 5.1 (Phase 1) 149
Figure 5.11: FNN Structure of Example 5.2 (Phase 2) 150
Figure 5.12: The Outline of The Proposed Approach 152
Figure 5.13: Accuracy (GM) of Different Baselines 167
Figure 5.14: Accuracy of Prediction for Year 2000 in Experiment A 170
Figure 5.15: Accuracy of Prediction for Year 1998 in Experiment A 171
Figure 5.16: Accuracy of Prediction for Year 1995 in Experiment A 171
Figure 5.17: Accuracy of Prediction for Year 1998 in Experiment B 173

FIGURE 5.18 : ACCURACY OF PREDICTION FOR YEAR 1998 IN EXPERIMENT B 174
FIGURE 5.19: ACCURACY OF PREDICTION FOR YEAR 1995 IN EXPERIMENT B 174
FIGURE 7.1: INTELLIGENT FINANCIAL WARNING SUPPORT SYSTEM FRAMEWORK 225
FIGURE 7.2: INTELLIGENT FINANCIAL WARNING SUPPORT SYSTEM MODEL 226

LIST OF TABLES

TABLE 2.1: DEFINITION OF ALL FINANCIAL RATIOS TO MEASURE CAMELS CRITERIA WHICH ARE USED IN THE LITERATURE OF BANK FAILURE PREDICTION 27

TABLE 2.2: DEFINITION OF FINANCIAL RATIOS AND THEIR IMPACTS ON BANK FAILURE .. 28

TABLE 2.3: CONFUSION MATRIX 50

TABLE 3.1: RESEARCH DESIGN 81

TABLE 3.2: NUMBER OF AVAILABLE RECORDS IN DATA SET 1 FOR EACH SCENARIO 81

TABLE 3.3: NUMBER OF AVAILABLE RECORDS IN DATA SET 2 FOR EACH SCENARIO 81

TABLE 3.4: RESULT OF PROPOSED APPROACH FOR 6 SCENARIOS, 2 DATA SETS AND 2 PRE-PROCESSING METHODS 85

TABLE 3.5: FRIEDMAN RANKING TO EVALUATE THE PREPROCESSING METHODS..... 85

TABLE 3.6: THE SIX CASES WHICH ARE USED FOR COMPARISON 86

TABLE 3.7: HOLM'S TEST FOR COMPARISON OF 3-VARIABLE SCENARIOS WITH 9-VARIABLE SCENARIOS..... 86

TABLE 3.8: ACCURACY OF FIVE PREDICTION MODELS AND PROPOSED APPROACH..... 90

TABLE 3.9: ACCURACY OF C4.5 WITH SMOTE-ENN AND PROPOSED APPROACH WITH SMOTE..... 91

TABLE 3.10: QUADE RANKING FOR ALL ALGORITHMS USING BALANCED DATA (SMOTE) 92

TABLE 3.11: HOLM TEST FOR COMPARISON OF ALL ALGORITHMS USING BALANCED DATA (MLP IS THE CONTROL MODEL, $\boldsymbol{\alpha = 0.05}$)..... 92

TABLE 3.12: HOLM TEST FOR COMPARISON OF ALL FNNS USING BALANCED DATA (THE PROPOSED APPROACH IS THE CONTROL MODEL, $\boldsymbol{\alpha = 0.05}$)..... 93

TABLE 3.13: RESULTS OF TWO PAIRED SAMPLE WILCOXON SIGNED RANK TEST FOR COMPARISON OF PROPOSED APPROACH WITH ANFIS AND MLP USING BALANCED DATA93

Table 3.14: Holm Test for Comparison of C4.5 Using SMOTE-ENN with Proposed Approach Using SMOTE 94
Table 3.15: Quade Ranking for All Algorithms Using Imbalanced Data 95
Table 3.16: Holm Test for Comparison of All Algorithms Using Imbalanced Data (The Proposed Approach Is The Control Model) 96
Table 3.17: The Accuracy of Proposed Approach when GM_error, RMSE and F-Measure Are Applied 99
Table 3.18: Holm Test to Compare The Accuracy Using GM_error and RMSE 100
Table 3.19: Fuzzy Rules Derived from The Fuzzy Neural Network in The Proposed Approach 102
Table 4.1: Similarity and Membership Value of a Set of Examples 111
Table 4.2: Dissimilarity and Membership Value of a Set of Examples 111
Table 4.3: Bank Records in Data Set 127
Table 4.4: Different Settings of Proposed Algorithm 128
Table 4.5: T-test Results to Examine Different Settings of Proposed Algorithm 131
Table 4.6: The Accuracy of MSFRB and MSBR Algorithms 132
Table 4.7: Holm Test for Comparison of MSFBR and MSFBR 132
Table 4.8: The Accuracy of Benchmarked Algorithms 133
Table 4.9: Holm Test for Comparison of Proposed Algorithm with TSVM_2BR, SVM_2BR and NB_2BR 134
Table 5.1: The FCC for Linguistic Terms of Example 5.1 153
Table 5.2: Synthetic Data Set 160
Table 5.3: Real World Financial (Bank Failure) Data Set 162
Table 5.4: Feature Weight and Baselines Accuracy (GM) 168
Table 5.5: Average Accuracy in Experiment A 170
Table 5.6: Average Accuracy in Experiment B 173
Table 5.7: Final Feature Weights for TraWei1&2 in Experiment A 175
Table 5.8: Final Feature Weights for TraWei1&2 in Experiment B 175

TABLE 5.9: HOLM TEST FOR COMPARISON OF TRAREFWEI1&2 WITH OTHER BASELINES *176*
TABLE 5.10: HOLM TEST FOR COMPARISON OF TRAREFWEI1&2 IN EXPERIMENTS A AND B *176*
TABLE 6.1: SETTING SPECIFICATIONS OF PROPOSED ALGORITHMS WHICH ARE USED IN COMPARISONS..... *205*
Table 6.2: Accuracy and Relative Increase in Accuracy Achieved by Refinement Algorithms on NB..... *208*
Table 6.3: Accuracy and Relative Increase in Accuracy Achieved by Refinement Algorithms on SVM..... *209*
Table 6.4: Accuracy and Relative Increase in Accuracy Achieved by Refinement Algorithms on MLP-NN..... *209*
Table 6.5: Accuracy and Relative Increase in Accuracy Achieved by Refinement Algorithms on TSVM..... *210*
Table 6.6: Accuracy and Relative Increase in Accuracy Achieved by Refinement Algorithms on FNN..... *210*
Table 6.7: Holm Tests (95% of Confidence) Examine The Influence of Fuzzy Approach on Refinement Algorithm Performance..... *212*
Table 6.8: Holm Tests (95% of Confidence) Examine The Influence of Similarity and Dissimilarity Functions on Refinement Algorithm Performance..... *212*
Table 6.9: Holm Tests (95% of Confidence) Examine The Influence of Multiple Steps on Refinement Algorithm Performance *213*
Table 6.10: Final Feature Weights when Different Prediction Models Are Applied *214*
Table 6.11: Accuracy and Relative Increase in Accuracy Achieved by Different Approaches when BN Is Prediction Model..... *215*
Table 6.12: Accuracy and Relative Increase in Accuracy Achieved by Different Approaches when SVM is Prediction Model *216*
Table 6.13: Accuracy and Relative Increase in Accuracy Achieved by Different Approaches when NN Is Prediction Model..... *216*

Table 6.14: Accuracy and Relative Increase in Accuracy Achieved by Different Approaches when TSVM Is Prediction Model *216*
Table 6.15: Accuracy and relative increase in accuracy achieved by different approaches when FNN is prediction model *217*
Table 6.16: Holm Tests (95% of Confidence) Comparison of FACDA Approach with MAC, MAB Approaches *217*

ABSTRACT

Financial early warning system aims to warn of the impending critical financial status of an organization. A financial early warning system is more than a classical prediction model and should provide an explanatory analysis to describe the reasons behind the failure; the explanatory ability of a system is as important as its predictive accuracy. In addition, failure prediction is intrinsically a class imbalance problem in which the number of failed cases is much less than the number of survived cases. Also, the vagueness in the value of predictors is an inevitable problem which has emerged in the uncertain environment of the finance industry. Scarcity of training data is another critical problem in finance industry; a new type of financial early warning system, which can be transferred and modified for different domains to transfer knowledge to new prediction domain, is highly desirable in practical applications because it is easy to install and cheap to setup.

To achieve the aforementioned properties, this study develops algorithms, methods and approaches in the case of bank failure prediction. First, a novel parametric adaptive inference-based fuzzy neural network approach is devised to predict financial status accurately and generate valuable knowledge for decision making. It handles the imbalance problem and the vagueness in features' value using parametric learning and rule generation algorithms. Second, a fuzzy domain adaptation method is developed to transfer knowledge from a related old problem to the problem under consideration and the labels are then predicted with a high level of accuracy. This method handles the data scarcity problem and enables the financial early warning system to be transferrable between prediction domains which are different in data distribution. Third, a fuzzy cross-domain adaptation approach is proposed to make the financial early warning system transferable from different but related domains to the current domain. This approach handles the problem in which the feature spaces of

prediction domains are different and have vague value. This approach selects the significant fuzzy predictors in the current prediction domain by transferring knowledge from the related prediction domains.

The proposed algorithms, methods and approaches are validated and benchmarked in each step of development using experiments performed on real world data. The results show that this study significantly enhances predictive accuracy at different stages of development. Finally a case study is performed to integrate and validate the proposed methods and approaches using Australian banking system data. The results demonstrate that this study successfully solves the abovementioned problems and significantly outperforms existing methods.

CHAPTER 1

INTRODUCTION

This chapter presents the introduction to this study. Section 1.1 provides the background to this research and the problem this study aims to solve. Sections 1.2 to 1.4 explain the challenges, objectives and contributions of this study. In Section 1.5, the research methodology, which is applied to conduct this study, is introduced. Section 1.6 describes the structure of this thesis and Section 1.7 addresses the publications related to this study.

1.1 BACKGROUND

Since the advent of various financial crises in the 1990s and 2000s-particularly the recent recession in mid-2008-there has been extensive investment in the construction of accurate computational systems to predict the probability of financial crises and bankruptcies. From 1980 to 1996 three-quarters of the International Monetary Fund (IMF) member countries experienced bank failures which were not restricted to particular geographic regions, levels of development or banking system structures (Davis & Karim 2008). These bank failures, along with many enterprise bankruptcies, which may have occurred due to radical changes in the global economy and customer demand for strong competition in uncertain operational environments, are given in previous research as clear evidence of serious distress. To tackle these problems, various data analysis models and prediction systems to forecast the financial situation of an organization- namely, financial early warning systems, have been developed.

Although it has been proved that these models are useful to managers and regulators with authority to prevent the occurrence of crises and failures (Ahn et al. 2000; Balcaen & Ooghe 2006), a number of drawbacks make them inapplicable as vital systems for business. (1) Most of the existing approaches, which use statistical methods, have deficiencies such as: ignoring important sources of uncertainty in classification as an arbitrary definition of failure; data instability and arbitrary choice of the optimization criteria; and neglecting the time dimension of failure (Balcaen & Ooghe 2006). (2) In addition, almost all existing statistical financial prediction models (Cole & Gunther 1995; Lane et al. 1986) have been criticized for their assumptions, which are more likely to be violated in the fields of finance and economics (Quek et al. 2009). (3) Similarly, they are not able to identify the traits of financial distress that lead to bank failure, and therefore function as black boxes (Tung et al. 2004). Conversely, the growing development of computational intelligence techniques has led researchers to employ new methods such as decision tree (Frydman et al. 1985), support vector machine (Ding et al. 2008; Hua et al. 2007; Min & Lee 2005; Shin et al. 2005), case based reasoning (Li & Sun 2010; Li & Ho 2009; Park & Han 2002), genetic algorithm (Shin & Lee 2002) and rough sets (Park & Han 2002) in financial early warning systems. Ravi Kumar and Ravi (2007) provide a detailed review of these models and methods in the domain of bankruptcy prediction and demonstrate their superior performance.

One of the most popular computational intelligence techniques that has been significantly applied to the domain of forecasting is neural network (Ahn et al. 2000; Fletcher & Goss 1993; Kim et al. 2004; Odom & Sharda 1990; Salchenberger et al. 1992; Wilson & Sharda 1994). A range of research results that apply various types of Neural Network for clustering, classification and prediction significantly improve accuracy in comparison with other methods. Although neural network is a well-known, efficient tool for prediction, it works as a ‘black box’ due to its computational framework. It learns only the relationship between inputs and outputs, without providing any knowledge about that relationship, which is critical for decision

making. Fuzzy systems have been introduced in this area to supply explanatory fuzzy rules describing the relationships among inputs and outputs. Fuzzy systems can tackle the imprecise nature of financial forecasting and effectively present expert knowledge about the influence of input variables on financial situation, as output through a fuzzy rule base (Alam et al. 2000; Lu et al. 2007; Tang & Chi 2005; Vigier & Terceño 2008). The ability of fuzzy systems to generate knowledge, and to use expert knowledge to solve prediction problems in an uncertain environment, makes them very popular in the financial domain but they are not as accurate as neural network.

The concept of integrating computational intelligence techniques to achieve a desirable result has been introduced in recent years and has led to the appearance of hybrid models. These models may embed different techniques in an integrated framework, or they may use different techniques separately, considering a unique weight for each one to generate a prediction. The result of these models in research has shown that they can out-perform other techniques with regard to some features in most cases (Ahn et al. 2000). For instance, fuzzy neural network, which is an embedded model, uses neural network and fuzzy systems to create a robust hybrid classifier and forecaster tool in different fields (Peymanfar et al. 2007; Sim et al. 2006a; Singh et al. 2008; Tung & Quek 2004). In recent research, different kinds of fuzzy neural networks have been used to classify and predict financial failures (Chen et al. 2009b; Lin et al. 2008; Ng et al. 2008; Quek et al. 2009; Tung et al. 2004). The main advantages of these models are their consistent fuzzy rule base gained from fuzzy systems along with their learning ability and accuracy obtained from neural network, to prevent probable future crises. These models not only predict the financial situation of a corporate business relatively accurately, but also provide a knowledge base which may be used to make decisions and prevent adverse circumstances from occurring and spiraling out of control. Although their knowledge generation ability makes them suitable for prediction, their prediction accuracy suffers in comparison with neural network, which is the most accurate prediction method, even with its limitation of functioning as a ‘black box’ (Ng et al. 2008).

The financial distress prediction problem, including bankruptcy and bank failure prediction, is inherently categorized as a class imbalance problem. This problem occurs when the number of instances of one class is much lower than the instances of other classes. The problem is extremely important, as it appears in many real-world applications (Haibo & Garcia 2009; Sun et al. 2009) such as failure prediction. Bank failure prediction particularly focuses on two class imbalanced data sets problems, where there is only one positive class (failed corporate) with a lower number of examples, and one negative class (survived corporate) with a higher number of examples. The problem appears in bank failure prediction because very few banks go failure in proportion and also it is difficult to obtain all the relevant data from the failed cases. Since this problem significantly affects the performance of the prediction model and reduces the accuracy, many studies have proposed methods to tackle this problem in various applications. Nevertheless, this problem has not been taken into consideration in bank failure literature when computational intelligence techniques are applied to form the prediction model.

Financial early warning systems, which have used machine learning technologies, have already achieved significant attention in research studies due to their accurate performance. However, almost all these methods work well only under a common assumption: namely, that the training and test data have identical feature spaces with underlying distribution. As a result, once the feature space or the feature distribution of the test data changes, the prediction models cannot be used and must be rebuilt and retrained from scratch using newly-collected training data, which is very expensive, if not practically impossible (Pan & Yang 2010). Similarly, since learning-based machine learning models need adequate labeled data for training, it is nearly impossible to establish a learning-based model for a domain (target domain) which has very few labeled data available for supervised learning. If we can transfer and exploit the knowledge from an existing related but not identical domain (source domain) with plenty of labeled data, however, we can pave the way for construction of the learning-based model for the target domain. In real world scenarios,

particularly in the finance industry, there are many situations in which very few labeled data are available, and collecting new labeled training data and forming a particular model are practically impossible. For instance, there are plenty of labeled data available for constructing a prediction model to specify bank status in the United States (source domain), whereas there are very few samples available for the banking system in Australia (target domain). Since they might not have identical feature spaces and distribution, it is not possible to use the model of United States banking system for Australian banks. However, they are similar and have common features, which may assist in the employment of the prediction model in the target domain.

Transfer learning has emerged in the machine learning literature as a means of transferring knowledge from a source domain to a target domain. Unlike traditional machine learning and semi-supervised algorithms (Blum & Mitchell 1998; Joachims 1999b; Nigam et al. 2000; Zhu 2005), transfer learning considers that the domains of the training data and the test data may be different (Fung et al. 2006). The study of transfer learning has been inspired by the fact that human beings can utilize previously-acquired knowledge to solve new but similar problems much more quickly. Research into transfer learning has been undertaken since 1995 under a variety of names: learning to learn, life-long learning, meta learning, and multi-task learning. Jialin and Qiang (2010) presented a comprehensive survey of transfer learning methods which introduced studies in transfer learning. Despite the recent surge of research in this field, certain issues have still not been taken into account and remain as challenges, such as handling the vagueness in feature values using soft computing methods, selecting significant features instead of instances in the target domain, and specifying an explicit relation among domains to construct a more general and independent model. Moreover, transfer learning, particularly domain adaptation, which is a new machine learning and data mining framework, can be implemented in many novel applications, but most studies have been conducted in text classification and reinforcement learning and there is a lack of published novel applications of transfer learning in other areas (Yang 2009).

To solve the abovementioned challenges and facilitate the transformation of current financial early warning systems into a new stage whereby the system achieves knowledge generation ability as well as high accuracy and the adaptation capability in the uncertain environment, this research proposes and develops a prediction model and transfer learning algorithms and approaches to overcome the limitations of existing financial early warning systems.

1.2 RESEARCH CHALLENGES

In this section, two main issues which significantly motivate the work presented in this thesis are reviewed.

(1) Majority of these systems have dealt with financial failure only as a classical prediction model. Creating a financial early warning system is one of the most interesting issues in Business Intelligence and has attracted many research efforts since 1960s. Based on the broad literature review, the need for an applicable financial early warning system which considers a range of important features along with accuracy, to aid managers in making critical decisions to save and guarantee their organizations against impending failure, is evident. To achieve an applicable system which can be accepted and utilized in the finance industry, it is necessary to consider the system as being more than a prediction model and because it should also provide an explanatory analysis to describe the reasons behind the financial failure. Financial failure is categorized as a class imbalance problem in an uncertain environment with huge databases which must be taken into account in constructing the financial early warning system. Thus, the first two questions of this research can be formulated as follows:

Research Question 1: How should the prediction model be formulated to gain the knowledge generation ability as well as a high rate of accuracy in the uncertain environment of the finance industry?

Research Question 2: How should the prediction model be formulated to handle the class imbalance problem efficiently to achieve a high level of accuracy when dealing with huge amounts of data?

(2) Machine learning-based financial failure prediction methods suffer from poor accuracy if the training and test data are not identical. Recently, many studies have applied machine learning methods to investigate financial failure prediction problems, including bank failure and bankruptcy. The reported results have demonstrated the significant effectiveness of these methods for failure prediction. However, these methods attain high performance if the training data and test data are extracted from identical sources, otherwise, they suffer from poor accuracy. More recently, transfer learning methods have been introduced to solve this problem. Although the published outputs of the proposed methods have revealed significant progress and promising performance, certain issues have still not been taken into account and remain as challenges:

(a) Most existing research in all categories of transfer learning uses probabilistic models which work well under statistical assumptions but may be violated in real world applications. Moreover, they are not able to tackle uncertain values of real world problems and consequently decline in performance. Fuzzy sets and rough sets are more flexible toward these assumptions and are capable of handling the uncertainty, but there is no study that uses these soft computing techniques for transfer learning.

(b) There is one category of transfer learning problem in which the distribution of training data and test data are different but the feature spaces are the same. Most of the existing transfer learning methods in this category, which is called domain adaptation, aim to refine the decision boundary or prediction models. This approach to solving problems makes these models highly complex computationally and dependent on the prediction model. Applying local learning and focusing on the given test data to refine the obtained labels would be an acceptable approach that is less computationally complex and more independent of the prediction model.

(c) Another category of transfer learning, which addresses the fact that the feature spaces of the training data and test data are different, is called cross-domain adaptation. Existing cross-domain adaptation methods focus on significant and similar instances in both domains to construct an implicit relation among domains and then transfer knowledge. Selecting significant features instead of instances in both domains will lead to an explicit relation among domains and will enable the construction of a more general and independent transfer learning model.

(d) Transfer learning, particularly domain and cross-domain adaptation, which are new machine learning and data mining frameworks, can be implemented in many novel applications, but most studies have been conducted in text classification and reinforcement learning and there is a lack of published novel applications of transfer learning in other areas such as financial early warning systems.

Accordingly, the remaining questions of this study can be formed as follows:

Research Question 3: How should the transfer learning method be formulated to efficiently handle the vagueness in feature values to achieve high level of accuracy when dealing with the uncertain environment of the finance industry, which may not follow predefined statistical assumptions?

Research Question 4: How should the domain adaptation method be formulated to use local learning instead of global learning and thus be more accurate and independent of a prediction model?

Research Question 5: How should the cross-domain adaptation method be formulated to focus on significant related features in both domains, instead of instances, to build an explicit relation among domains and thus construct a more general and independent method?

Research Question 6: How effective and accurate is the transfer learning approach, including domain adaptation and cross-domain adaptation methods, when it is applied to more general applications such as the bank failure prediction model?

1.3 RESEARCH OBJECTIVES

In view of the research challenges introduced above, a prediction approach in addition to a domain adaptation method and cross-domain adaptation approach should be designed and developed. The main objectives and significance of this thesis are:

Research Objective 1: To develop a prediction approach with a high level of accuracy, explanatory ability and capability to handle the class imbalance problem.

This objective corresponds to Research Questions 1 and 2. To overcome the shortcomings of classical prediction methods, which only emphasize predictive accuracy, a fuzzy neural network as a hybrid prediction model will be developed. This model not only predicts the financial status competitively accurately but also provides a fuzzy rule base as valuable knowledge for decision making. The input of this approach is the value of financial ratios of the underlying organization. The outputs are the predicted financial status of the company and fuzzy rule base that explains the reasons for this prediction. The proposed model is able to handle the class imbalance problem through a preprocessing step and a novel learning algorithm which is appropriate for dealing with large amounts of data.

Research Objective 2: To develop a fuzzy transfer learning approach to make the early financial warning system transferable in the uncertain environment of business finance.

This objective corresponds to Research Question 3. As mentioned in Objective 1, the proposed financial early warning system uses fuzzy neural network which is a machine learning technique. In this technique, the data must come from the same feature representation and distribution in the training and test data. However, in many real-world applications, this assumption does not hold. For example, there is a prediction model in one domain (United States banking system), but there are insufficient training data in another domain (Australian banking system) where the data may be in a different feature space or may follow a different distribution. In these cases, knowledge transfer, if done successfully, would greatly benefit the learning in our interested domain by avoiding expensive training tasks. Since the

financial data may be vague, the proposed approach handles the failure prediction problem with uncertain values in features by using a fuzzy approach for transferring knowledge. This part of the study focuses on developing a fuzzy transfer learning approach, including domain and cross-domain adaptation methods to cope with the fuzzyness in features and thus gain high level of accuracy.

Research Objective 3: To develop a domain adaptation method that uses local learning for transferring the knowledge between domains.

This objective corresponds to Research Question 4. This method is the first to utilize fuzzy set techniques to handle vague values of instance features in domain adaptation. Instead of modifying the baseline model or decision boundary, this method introduces a fuzzy similarity/dissimilarity–based learning method as a local learning for domain adaptation. This study explores similar/dissimilar fuzzy instances in the bridged domains and then, using the explored instances, refines the pseudo labels in the test data sets that were initially established by a prediction model like fuzzy neural network. The novel domain adaptation method refines the predicted labels and focuses on currently given test data instead of modifying the decision boundary, which makes the algorithm more independent of the prediction model. In particular, the algorithm applies multi-step label refinements in mixture domains towards target distribution to halt the influence of the prediction model on performance. The proposed label refinement is performed simultaneously, based on the similarity and dissimilarity of mixture domains instances. These abilities enable the proposed method to be more accurate and capable of being practically implemented in real world applications, particularly in financial businesses with huge databases.

Research Objective 4: To develop a fuzzy cross-domain adaptation approach to explore significant and related features in both domains and thus establish an explicit relation between domains to transfer knowledge.

This objective corresponds to Research Question 5. This approach bridges the gap between source and target domains by aligning domain-specific features with the help of domain-independent features and selecting the significant features in the target

domain. It explicitly represents the relationship between the two domains by depicting the correlation between the domain-specific features of both domains through the domain-independent features. It specifies the significant features instead of instances in the target domain. The features are selected based on two weights achieved from domain-Independent features which are similar in both domains, and on domain-Specific features which are different but have significant correlation. Compared with existing models, the proposed approach is more flexible toward the assumptions of probabilistic models and is able to handle the vagueness of feature values. In particular, by modifying the predicted labels in the domain-independent space and co-clustering features in the domain-specific space, the approach solves both distribution difference and feature space difference problems at the same time. The approach focuses on currently given training and test data rather than the baseline model and decision boundary, and is thus more general and independent of the prediction model. It is universal, such that it can be applied to bank failure prediction and is not limited to particular applications such as natural language processing and text classification – a point which is worth emphasizing.

Research Objective 5: To conduct a case study of bank failure prediction to evaluate the effectiveness of the proposed methods through an integrated approach using real world financial data from two domains.

This object corresponds to Research Question 6. This case study will be performed based on the proposed prediction model, domain adaptation and cross-domain adaptation methods and approaches. The financial data are extracted from United States and Australian banking systems as source and target domains respectively to conduct the case study. The main function of the case study is to integrate and evaluate proposed methods, algorithms and approaches, which have been validated separately by empirical analysis, in a specified approach. The case study represents the proposed financial early warning system as an integrated approach. It measures the accuracy of the proposed system, evaluates the achieved increase and compares it with existing approaches. This is the first study to apply a transfer learning method to

a real world financial application such as bank failure prediction and to exploit the knowledge of the banking system of one country to establish a prediction model in another country.

1.4 RESEARCH CONTRIBUTIONS

The financial early warning system, also known as bankruptcy prediction, business failure and bank failure prediction, is one of the most interesting research fields in information technology and finance. The explanatory nature, generality and transferability of these systems that make them efficient and practical in the finance industry is a novel research direction in this domain. Based on the above objectives, the expected research contributions can be summarized as follows:

(1) This study develops a new fuzzy neural network as a prediction model. It predicts the financial situation more accurately than existing methods and also generates knowledge in the form of a fuzzy rule base to explore the reasons behind the prediction. The results of this study will make a remarkable contribution to the prediction models research field by proposing a novel fuzzy neural network. The proposed prediction model should be significantly accurate and should be able to generate knowledge. The model uses an adaptive inference-based learning algorithm which modifies critical parameters in the inference system to obtain the best result. Compared to other methods which change the fuzzy terms in the database for training, the proposed model does not change anything in the database and consequently will be faster, cheaper and more practical in real cases. Likewise, the proposed model includes preprocessing steps which reduce the class imbalance problem's negative influence and brings about a more accurate result. These abilities make the system more applicable to the business finance situation.

(2) This study proposes innovative the fuzzy domain adaptation method. The method has great value in transfer learning studies by virtue of proposing a new domain adaption method. Transfer learning is a new research direction in machine learning which has only recently begun to gain much attention. In this method, the knowledge

from another domain, which is different but similar to the domain under consideration, is used to improve the learning performance. This study proposes a novel fuzzy domain adaption method that can cope with uncertainty issue in feature values using fuzzy techniques. Also, it uses a local learning approach to transfer the knowledge between domains by focusing on given data. These achievements will make a remarkable contribution to transfer learning research studies.

(3) The design and development of a fuzzy cross-domain adaptation approach. Most existing studies in transfer learning focus on the domain adaptation problem and few researches have investigated the cross-domain adaptation problem. Even the few existing cross-domain adaptation studies have only focused on the problems by using probabilistic models with crisp values. The proposed approach applies fuzzy techniques and heuristic models to handle the instances with vague values. It focuses on exploring the significant features in the target domain according to the knowledge of the source domain. It transfers and adjusts the feature representation of domains instead of concentrating on samples. Using this approach, it builds an explicit relation among domains that can clearly interpret the cross-domain adaptation function. These achievements make a valuable contribution to transfer learning research study.

(4) The design and conduct of a case study for a financial early warning system for Australian banks using United States banking system data. Transfer learning has been introduced in limited real world applications; it has not been applied to a range of more general applications such as bank failure prediction. Implementing this method in financial failure prediction introduces a valuable new element to this field. This study aims to integrate the proposed methods into an approach which is then examined through a case study of Australian banks. This case study is an innovative example of applying machine learning and transfer learning to a financial early warning system. It significantly contributes to the financial failure forecasting literature, and to bank failure prediction in particular.

(5) The proposed approach notably enhances the performance and improves the quality of current financial early warning systems, and upgrades them to be more

widely usable in the finance industry. The proposed approach has a high level of predictive accuracy and knowledge generation ability in prediction due to the development of the fuzzy neural network. It is more flexible, cheaper, faster and more practical because it uses a transfer learning method. The proposed approach can first be trained (installed) using an accessible, appropriate database in a domain, and can then be trained and used in other domains which may have insufficient data by transferring the knowledge from the first domain. For instance, the system can be installed for a banking system in one state that has data with an extensive history: this same system can subsequently be installed in a newly-established banking system in another state, using the knowledge gained previously.

(6) The proposed approach will significantly help industry to reduce the financial risk of individual organizations and will assist them to be safe and profitable. The financial early warning system is an important and serious topic for business because effective prediction and in-time decision making is invaluable in the prevention of failure. The proposed approach assists organizations to discover their financial weaknesses and problems and to then predict their status accurately and in time, based on their current status. The descriptive prediction output aids managers not only to prevent probable failure and keep the company secure, but also to lead the organization into a more stable and prominent situation. Concisely, this approach paves the way for organization to guarantee that its financial future will be profitable.

1.5 Research Methodology and Process

Research methodology is the "collections of problem solving methods governed by a set of principles and a common philosophy for solving targeted problems"(Gallupe 2007). A number of research methodologies have been proposed and applied in the Information System domain such as case study, field study, design research, field experiment, laboratory experiment, survey, and action research (Vaishnavi & Kuechler 2009).

1.5.1 RESEARCH METHODOLOGY

In this research, the design research is considered to be the most appropriate research methodology to achieve the research objectives (Niu et al. 2009). The methodology, as illustrated in Figure 1.1, includes five basic stages.

1.5.1.1 AWARENESS OF PROBLEM

This is the first step, in which the limitations of existing applications are analyzed and significant research problems are acknowledged. The research problems reflect a gap between existing applications and the expected status. Research problems can be identified from different sources: industry experience, observations on practical applications and literature review. A clear definition of the research problem provides a focus for the research throughout the development process. The output of this phase is a research proposal for new research effort (Niu et al. 2009; Vaishnavi & Kuechler 2009).

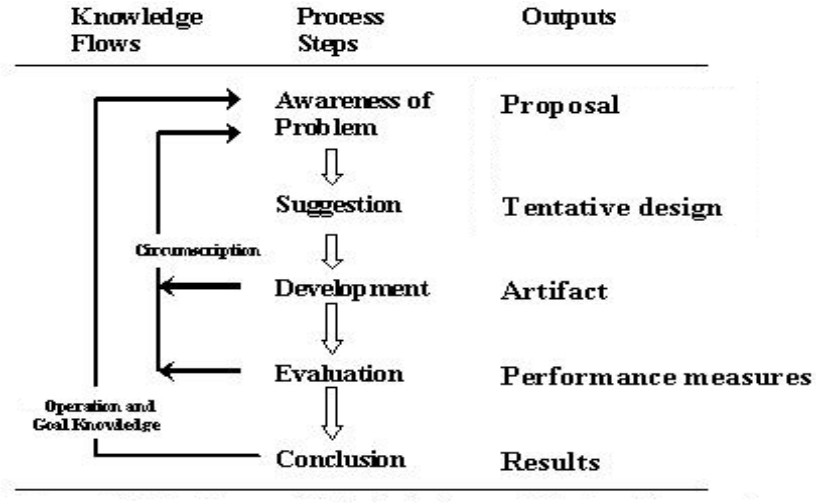


FIGURE 1.1: THE GENERAL METHODOLOGY OF DESIGN RESEARCH (NIU ET AL. 2009)

1.5.1.2 SUGGESTION

This phase immediately follows behind the identification of research problems and a tentative design is suggested. The tentative design describes what the prospective

artifacts will be and how they can be developed. Suggestion is a creative process during which new concepts, models and functions of artifacts are demonstrated. The resulting tentative design of this step is usually one part of the research proposal; thus, the output of the suggestion step is also feedback of Step (1), whereby the research proposal can be revised (Niu et al. 2009; Vaishnavi & Kuechler 2009).

1.5.1.3 DEVELOPMENT

This phase considers the implementation of the suggested tentative design artifacts. The techniques for implementation will be based on the artifact to be constructed. The implementation itself can be simple and need not involve novelty; the novelty is primarily in the design, not the construction of the artifact. The development process is often an iterative process in which an initial prototype is first built and then evolves as the researcher gains deeper comprehension of the research problems (Niu et al. 2009; Vaishnavi & Kuechler 2009).

1.5.1.4 EVALUATION

This phase considers the evaluation of the implemented artifacts. Artifact performance can be evaluated according to criteria defined in the research proposal and the suggested design. The evaluation results, which may or may not meet expectations, are fed back to the first two steps. Accordingly, the proposal and design might be revised and the artifacts might be improved (Niu et al. 2009; Vaishnavi & Kuechler 2009).

1.5.1.5 CONCLUSION

This is the final phase of a design research effort. Typically, it is the result of satisfying with the evaluation results of the developed artifacts. Though there are still deviations in the behavior between the proposal and the artifacts that are actually developed, a design research effort concludes as long as the developed artifacts are

considered to be ‘good enough’. Any anomalous behavior may well serve as the subject of further research (Niu et al. 2009; Vaishnavi & Kuechler 2009).

1.5.2 RESEARCH PROCESS

This research was planned according to the methodology of design research. First, a subject was chosen as a very broad research topic of this research. A literature review of previous research in the topic area is an essential component of the research process, so existing literature was retrieved and critically reviewed. The results of the literature review helped to define specific research questions to be directly addressed in the research project. As the research questions grew clearer and more definite, more literature closely related to the research questions was reviewed. Based on existing work in the literature, a set of novel models, algorithms and approaches were designed, developed and evaluated. The proposed models and algorithms were implemented and evaluated within the Matlab programming environment. According to the methodology of design research, this research is an iterative process. As indicated in Figure 1.1, the output of each research step might be fed back to its previous step when deviations between expectations and evaluation results are found. Through the feedback, research outcomes are progressively improved until satisfying results are drawn from evaluations. Finally, writing up the PhD thesis is done at the end of the research.

1.6 THESIS STRUCTURE

This thesis contains seven chapters. Chapter 1 presents the research background, challenges, objectives and significance, contributions, methodology, and thesis structure. Chapter 2 presents the literature relevant to this study, including a review of financial early warning systems and bank failure prediction models, the class imbalance problem and solutions, and transfer learning methods. Chapter 3 introduces a novel fuzzy neural network model including rule generation and learning algorithms, to handle the class imbalance problem. Chapter 4 describes a fuzzy domain

adaptation method including three similar algorithms, called multi-step fuzzy bridge refinement (MSFBR), to solve the long term prediction problem in which the feature spaces of both domains are the same but the distributions of data are different. The proposed method refines the initial labels, which were predicted by the prediction model (Chapter 3), based on the similarity and dissimilarity of instances in a number of mixture domains to achieve a more accurate result. Chapter 5 introduces a fuzzy cross-domain adaptation approach, called feature alignment-based cross domain adaptation (FACDA), which includes five main phases for solving the transfer learning problem in which the feature spaces of two domains are different. In Phase One of the proposed approach, the initial labels are computed by the prediction model (Chapter 3). In Phase Two, the MSFBR method (Chapter 4) is applied to refine the predicted labels on domain independent feature space. Phases Three to Five aim to find the most significant features in the target domain. Chapter 6 presents the case study to implement the proposed algorithms, methods and approaches. The case study investigates two different problem settings to which this research applies all the proposed algorithms to seek a solution. Chapter 7 presents the conclusions and future research directions for the work presented in this thesis. The structure of the thesis is clearly shown in Figure 1.2.

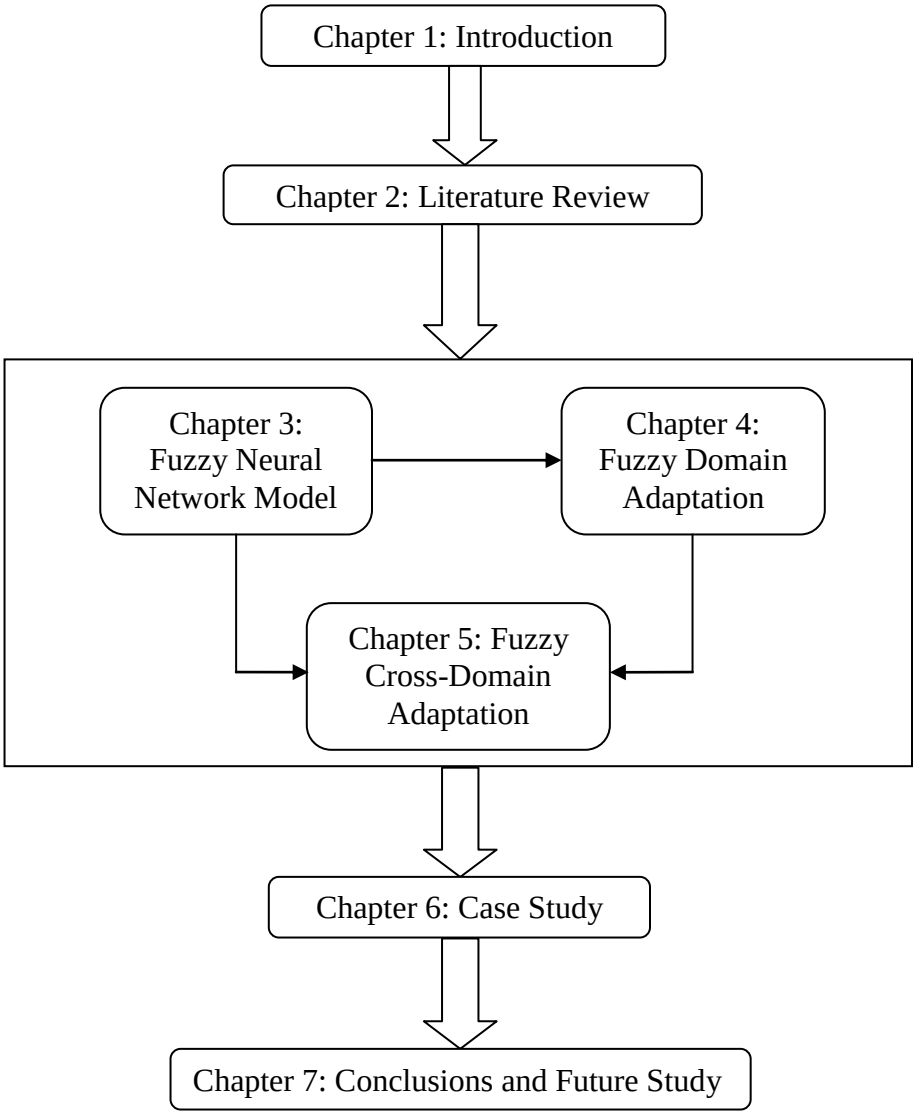


FIGURE 1.2: THESIS STRUCTURE

1.7 PUBLICATIONS RELATED TO THIS THESIS

Below is a list of the international refereed journal and conference papers associated with my PhD research that have been submitted, accepted and published:

Refereed International Journal Publications:

(1) V. Behbood, J. Lu, G. Zhang**, "*Adaptive Inference-based Learning and Rule Generation Algorithms in Fuzzy Neural Network for Failure Prediction,"*** Neural Networks, 2012 (Under Review)

(2) V. Behbood, J. Lu, G. Zhang**,** "***Fuzzy Cross-Domain Adaptation: Inter-State Bank Failure Prediction",*** Machine Learning Research, 2012 (Under second Review)

(3) V. Behbood, J. Lu, G. Zhang**, "*Multi-Step Fuzzy Bridged Refinement Algorithm: Long Term Bank Failure Prediction",*** IEEE Transactions on Data and Knowledge Engineering, 2012, (Under second review)

(4) V. Behbood, J. Lu, G. Zhang, ***"Fuzzy Refinement Domain Adaptation for Long Term Prediction in Banking Ecosystem"***, Accepted by IEEE Transactions on Industrial Informatics, 2012.

(5) V. Behbood, J. Lu, G. Zhang & W. Pedrycz, " ***Multi-Step Fuzzy Bridged Refinement Domain Adaptation Algorithm And Its Application To Bank Failure Prediction",*** IEEE Transaction on Fuzzy System, 2012, (Submitted)

(6) V. Behbood, J. Lu, G. Zhang, "***Fuzzy Domain Adaptation: A solution for data shortage in Bank Failure Prediction",*** Decision Support System, 2012, (Under review)

Refereed International Conference Publications:

(7) V. Behbood, J. Lu, G. Zhang**, *"Adaptive Inference-based Learning and Rule Generation Algorithms in Fuzzy Neural Network for Failure Prediction,"*** International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 33-38, 15-16 Nov. 2010, China.

(8) V. Behbood, J. Lu, G. Zhang and S.X. Wei, ***"Intelligent Financial Warning Support System,"*** International Conference on Applied Statistics and Financial Mathematic, 16-18 Dec. 2010, Hong Kong.

(9) J. Lu, G. Zhang, Y. Gao, J. Zhang, V. Behbood, "***Decision Support and Warning Systems for Business Intelligence*****,"** 10th International Conference on Information (ICI10), 4-6 Dec. 2010, Egypt.

(10) V. Behbood, J. Lu, **"*Intelligent Financial Warning Model Using Fuzzy Neural Network and Case-Based Reasoning,"*** IEEE Symposium on Computational Intelligence for Financial Engineering & Economics, 11-15 Apr. 2011, France.

(11) V. Behbood, J. Lu, G.Zhang, **"*Fuzzy Refinement-based Transductive Transfer Learning for Bank Failure Prediction,"*** World Conference on Soft Computing, 23-26 May. 2011, United States.

(12) V. Behbood, J. Lu, **"*Efficiency Prediction in Decision Making Units Merger using Data Envelopment Analysis and Neural Network,"*** 19th Triennial Conference of the International Federation of Operational Research Societies, 10-15 Jul. 2010, Australia.

(13) V. Behbood, J. Lu, ***"Financial Early Warning System: Adaptive Inference-based Fuzzy Neural Network,"*** 19th Triennial Conference of the International Federation of Operational Research Societies, 10-15 Jul. 2010, Australia.

(14) Behbood, V., Lu, J. and Zhang, G. (2011), ***"Long Term Bank Failure Prediction using Fuzzy Refinement-based Transductive Transfer Learning"***, IEEE International Conference on Fuzzy Systems, 27-30 Jun. 2011, Taiwan.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a discussion of the research background and relevant works in connection with this research. In Section 2.1, an overview of studies conducted into financial early warning systems and bank failure prediction in particular is provided. Section 2.2 addresses various significant features and ratios in existing studies. Sections 2.3 and 2.4 review two main categories of models: statistical and intelligent methods, employed to implement the systems. Section 2.5 introduces the imbalance problem raised in data sets and provides an explanation of existing solutions and accuracy measurements to solve the problem. Section 2.6 explains current state-of-the-art transfer learning techniques.

2.1 BANK FAILURE PREDICTION AND EARLY WARNING SYSTEM

From 1980 to 1996, three-quarters of the International Monetary Fund (IMF) member countries experienced bank failures which were not restricted to particular geographic regions, levels of development or banking system structures (Davis & Karim 2008). These bank failures, along with many enterprise bankruptcies, which may have occurred due to radical changes in the global economy and customer demand for strong competition in uncertain operational environments, have been cited in previous research as clear evidence of serious distress. To tackle these problems, various data

analysis models and prediction systems, called Financial Early Warning Systems (FEWS), have been developed. FEWS as a financial prediction model has been extensively researched since the late 1960's (Altman 1968). In previous researches, FEWS has been referred to many times by other terms such as *"Business Failure Prediction"*, *"Bankruptcy Prediction"*, *"Distress Prediction"* and *"Bank Failure Prediction"*, which interestingly shows that this system has been solely considered as a prediction model rather than a comprehensive decision support system. Nowadays, many voices have called for a revolution of existing FEWS to detect probable financial failure and take appropriate action to prevent bankruptcy problems (Chen et al. 2009b). Most creditors, auditors and senior managers are interested in a FEWS which allows them to monitor financial performance and identify the reasons for problems. In particular, since the advent of various financial crises in the 1990s and 2000s, especially the recent recession in mid-2008, there have been extensive investments in the construction of accurate computational systems to predict the probability of financial crises and bankruptcies.

The bank failure prediction model, which is one of the main types of FEWS, has attracted significant research attention since the 1970s when challenges arose in bank management (Huang et al. 2012). Most central banks have employed various FEWS to monitor the risk of banks for years; however, the repeated occurrence of banking crises during the past two decades — such as the Asian crisis, the Russian bank crisis, the Brazilian bank crisis, and particularly the recent US and UK bank crises — indicates that safeguarding the banking system is no easy task. The financial sector of the world will remember the year 2008 as one of the most shocking periods in history. After many solid years of growth, banks started to fail with increasing speed. Figures 2.1 and 2.2 clearly demonstrate the critical situation in which the banking industry found itself. There were no commercial bank failures in 2005 or 2006 and only three failures in 2007. In 2008, however, 25 financial institutions failed, giving a foretaste of the 140 and 157 failures in 2009 and 2010 respectively. The total estimated loss dramatically climbed to 40 billion dollar in 2010 from zero dollars in 2007. This

condition has rekindled interest in creating an accurate FEWS for bank failure. With an efficient FEWS, banks with financial difficulties can be identified, giving bank regulators sufficient time and information to react prior to the problem getting out of control. FEWS are explanatory prediction tools for individual bank failures or for detecting the financial distress of a complete banking system. In this study, the focus is on individual bank failure, not on discovering the distress level or depth of the whole banking sector. Therefore, every time FEWS is referred to, a failed/survived binary event is indicated, in contrast to the index of banking system distress that can take a continuum of values.

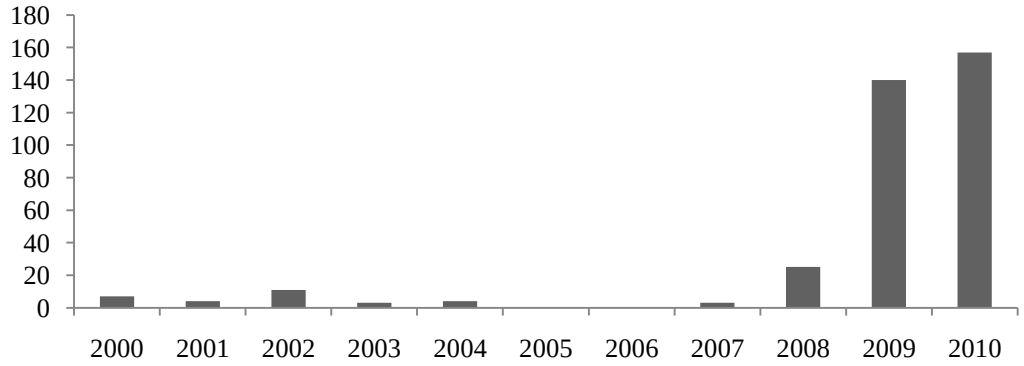


FIGURE 2.1: NUMBER OF FAILED BANKS IN UNITED STATES

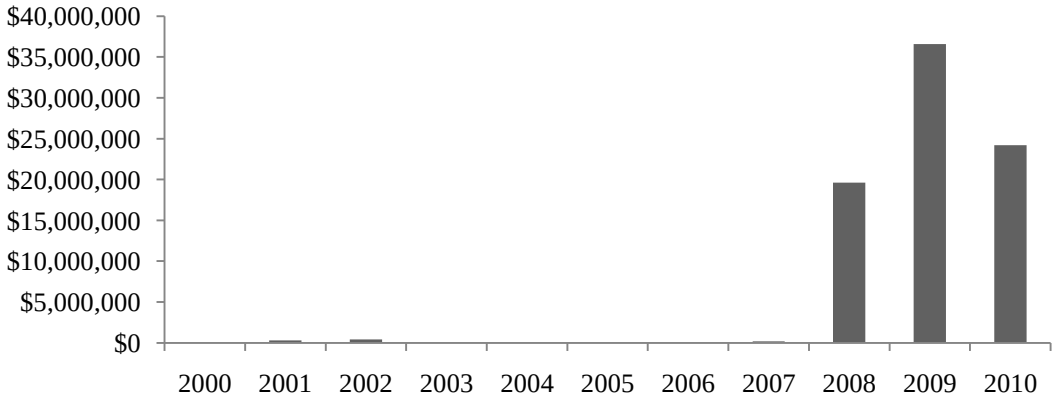


FIGURE 2.2: TOTAL ESTIMATED LOST (THOUSANDS DOLLAR)

FEWS can be divided into two categories: (1) on-site assessments, and (2) off-site assessments (Cole & Gunther 1998). On-site assessment is conducted on the premises

of a bank, by examining bookkeeping records, business books, subsidiary ledgers, and other records and accounts to evaluate the bank's financial soundness and compliance with laws and regulatory policies. Off-site analysis, on the other hand, can be carried out using publicly available financial information only. This information includes annual and quarterly reports that banks are obliged to compile for the regulators. Although on-site assessment is inclusive and precise, off-site analysis takes less effort and can be done frequently, which makes it a valuable tool for regulators. Cole and Gunther (1998) argued that off-site examination can even be more accurate than on-site assessment. The positive aspects of off-site examination, given the difficulty of visiting numerous banks across the country, make off-site models much more appropriate. This study aims to create an off-site FEWS for bank failure which analyses individual banks based on public financial statements.

2.2 FEATURE SELECTION

Selecting a set of comprehensive and effective features which significantly contribute to financial failure and also have minimum overlap is the first step in creating a FEWS. A group of researchers (Bhargava et al. 1998; Laitinen & Laitinen 2000) focused on financial data and variables obtained from cash flow. Other researchers (Atiya 2001; Becchetti & Sierra 2003; Donoher 2004; Fabling & Grimes) had great interest in non-financial factors such as macro-economic management, corporate management and even stock price volatility. The financial health of a corporate is generally dependent upon four main high level factors (Ravi Kumar & Ravi 2007): (1) how financially solvent it is at the inception; (2) its ability, relative flexibility and efficiency in creating cash from its continuous operations; (3) its access to capital markets; (4) its financial capacity and staying power when faced with unplanned cash short-falls. Although selecting variables or factors that play a prime role in constructing FEWS is a disputable issue, financial ratios are irreplaceable because of their long history in bankruptcy research (Chen et al. 2009b).

To find the most significant variables which can provide a widespread interpretation to prediction and decision making, all the important and influential financial factors that cause failures should be taken into account. As previously mentioned, a precise way of monitoring banks is to conduct on-site examinations. According to the Federal Deposit Insurance Corporation Improvement Act of 1991, regulators in the United States must conduct on-site examinations of bank risk every 12-18 months. Regulators use a rating system called CAMELS to indicate the safety and soundness of banks. This ranking system evaluates an organization according to six financial ratios: (1) Capital adequacy; (2) Asset quality; (3) Management expertise; (4) Earning strength; (5) Liquidity; (6) Sensitivity to market risk. The off-site financial ratios, which are calculated based on the CAMELS ranking system, are the most commonly used variables and can forecast potential failures rather well (Quek et al. 2009; Ravi Kumar & Ravi 2007). A comprehensive review of bank failure prediction literature has revealed that many researches have applied the explanatory financial ratios constructed to measure the six CAMEL components for bank failure prediction. As a collected reference for bank failure prediction, there are thirty-two different financial ratios that can be categorized in seven categories (FDIC 2008)[1] as shown in Table 2.1. Since feature selection is not part of this research scope, nine financial ratios are selected among above variables which are the most popular and acceptable financial ratios applied in the literature of bank failure prediction. The definition of these financial ratios, which are extracted from Balance-Sheet and Call Reports provided by Federal Deposit Insurance Corporation (FDIC 2009)[2], are described in Table 2.2.

[1] http://www2.fdic.gov/hsob/SelectRpt.asp?EntryTyp=30
[2] http://www2.fdic.gov/sdi/main.asp

TABLE 2.1: DEFINITION OF ALL FINANCIAL RATIOS TO MEASURE CAMELS CRITERIA WHICH ARE USED IN THE LITERATURE OF BANK FAILURE PREDICTION

CAMELS Criteria	Financial Ratios	Definition
Liquidity		
1)	Liquid assets	Total security holdings/Total assets
2)	Uninsured deposits	Time deposits of 100,000 or more/Total time deposits
Credit Risk		
3)	Loan Exposure	Total Loans and leases/Total assets
4)	Loan funding	Total loans and leases/Total deposits
5)	Nonaccrual rate	Assets in nonaccrual status/Total assets
6)	Past due loan rate	Assets Past due 90 or more days/Total assets
7)	**Loan loss allowance**	**Loss allowance/Total Loans and leases**
8)	***Provision rate**	**Loan loss provision /Total loans and leases**
9)	Loss rate	Net charge-offs/Total loans and leases
10)	***Capital ratio**	**Total equity capital/Total assets**
Profitability and Taxes		
11)	Return on assets	Net income/Total assets
12)	**Return on equity**	**Net income/Total equity capital**
13)	Dividend rate	Cash dividends/Total assets
14)	**Net interest margin**	**Net interest income/Total assets**
15)	Net operating margin	Net operating income/Total assets
16)	Tax exposure	Applicable income taxes/Total assets
Growth		
17)	Capital growth	Total equity (t)-Total equity (t-1) / Total equity
18)	**Loan growth**	**Total loans and lease (t) –Total loans and leases (t-1)/Total loans and leases (t)**
Loan and Deposit Mix		
19)	Commercial loan risk	Commercial and industrial loans/Total loans and leases
20)	Real estate loan risk	Total real estate loans/Total loans and leases
21)	Agricultural loan risk	Total agricultural loans/Total loans and leases
22)	Credit card loan risk	Credit card loans/Total loans and leases
23)	Loan diversification	Sum of squared proportions of four loan mix for each bank
24)	Demand deposit mix	Demand deposit/Total deposits
25)	Time deposits mix	Time deposits/Total deposits
Securities		
26)	MBS ratio	Mortgage-backed securities /Total securities
27)	ABS ratio	Asset-backed securities/Total securities
28)	CMO ratio	Collateralized mortgage obligation/Total securities
29)	Risk free securities	Government dept securities/Total securities
Instability		
30)	Assets variation	Total assets/Mean of total loans and leases
31)	Loans and leases variation	Total loans and leases/Mean of total loans and leases
32)	Equity variation	Total equity/Mean of equity capital

TABLE 2.2: DEFINITION OF FINANCIAL RATIOS AND THEIR IMPACTS ON BANK FAILURE

CAMELS Criteria	Financial Ratios	Definition
Capital adequacy		
	1) * Capital ratio	**Total equity capital / Total assets**
Asset quality		
	2) Loan loss allowance	**Loss allowance/ Total loans and leases**
	3) Past due loan measure	Average Loans 90+days late /Total loans and leases
	4) *Provision rate	**Loan loss provision /Total loans and leases**
Management expertise		
	5) Non-interest profit	Non-interest expense /Operating income
Earnings		
	6) Net interest margin	**Net interest income /Total assets**
	7) *Return on equity	**Net income /Total equity capital**
Liquidity		
	8) Liquidity measure	Cash + Federal funds sold/Total deposit + Federal funds purchased + Banks' liability on acceptance + Other liabilities
Miscellaneous		
	9) Loan growth	**Total loans and leases (t) - Total loans and leases (t-1) / Total loans and leases (t)**

2.3 FINANCIAL FAILURE PREDICTION: CLASSICAL STATISTICAL METHODS

Over the last five decades, the topic of business failure prediction and particularly bank failure prediction has developed into a major research domain within the financial industry. Many academic studies have been dedicated to finding the best failure prediction model. Academic researchers throughout the world have used various modeling techniques, each having distinct assumptions and specific computational complexities, to accurately classify and predict banks' status according

to their financial health. The most commonly used methods are the statistical methods, which have resulted in numerous static failure prediction models. The existing statistical models used as FEWS can be categorized into four main groups (Balcaen & Ooghe 2006):

(1) Univariate analysis, which was proposed by Beaver (1967) for the first time in 1966, is the simplest method which assumes a linear relationship between all financial measures and the failure status. In a univariate failure prediction model, an optimal cut-off point is estimated for each financial ratio in the model and a classification procedure is carried out separately for each ratio, based on an institution's value for the ratio and the corresponding optimal cut-off point. Even though univariate modeling technique is extremely simple and the application does not require any statistical knowledge, it is based on the strict assumption of a linear relationship between all financial ratios and the failure status.

(2) Risk index analysis, which was first proposed by Tamari (1966) and followed by Moses and Liao (1987) in 1966 and 1987 respectively, is an intuitive point system. This first model uses various weighted ratios to allocate a point which indicates the financial situation of an organization. A higher total point indicates a better financial situation. More significant ratios have larger weights, though the weights are allocated subjectively. The second model applies univariate analysis to compute cut-off points for each composing ratio. A binary variable is allocated to each ratio and if the ratio value exceeds the optimal cut-off point, the binary variable gains the value one. Finally the organization's financial status is specified by accumulating the binary scores.

(3) Multivariate Discriminate Analysis (MDA), proposed by Altman (1968) in 1968, consists of a linear combination of independent financial ratios which classifies the corporations as failing or non-failing. Until the 1980s, the MDA technique dominated the literature on business failure prediction. The majority of MDA studies used a linear MDA model, but quadratic MDA has also been applied (Balcaen & Ooghe 2006). Using the MDA method, the organizations are classified on the basis of their

discriminator score and the optimal cut-off point. If their discriminator scores are less than the cut-off point, they are classified as failed, whereas if their score exceeds or equals the cut-off point, they are classified as survived. The MDA model is based on three restrictive assumptions: (a) multivariate normally distributed independent variables; (b) equal variance-covariance matrices across the failed and survived group; and (c) specified prior probability of failure and misclassification costs. Since these assumptions are difficult to satisfy in financial data, most MDA failure prediction studies do not check whether the data satisfy the assumptions or not. As a result, the MDA model is not suited for generalization and its application has decreased since the 1980s. However, it remains a generally accepted standard method and is frequently used as a baseline method for comparative studies (Aziz & Dar 2006).

(4) Conditional probability models include the *LOGIT* model which assumes a logistic distribution (Hosmer & Lemeshow 1989; Maddala 1977; Martin 1977) and the *PROBIT* model which assumes a cumulative normal distribution (Theil 1971) for independent variables. LOGIT, which is more popular, is a non-linear maximum likelihood estimation procedure. It is used to obtain the probability of failure given the vector of financial ratios. It specifies a given organization as failed or survived based on its LOGIT score and a certain cut-off score for the model. If the LOGIT score is more than the cut-off score then the organization is classified as survived and vice versa. Compared to the MDA, since LOGIT does not require multivariate normal distributed variables assumption, it can be considered less demanding than MDA. However, Mcleay and Omar (2000) reported that LOGIT remains sensitive to extreme non-normality and is extremely sensitive to multicollinearity.

Canbas et al. (2005) proposed an Integrated Early Warning System (IEWS) that combines DA, LOGIT, PROBIT, and Principal Component Analysis (PCA), which can help predict bank failure. First, they applied PCA to identify three financial stages explaining the changes in the financial condition of banks. They then integrated DA, LOGIT and PROBIT regression models to construct an IEWS. The authors use the data for 40 privately owned Turkish commercial banks to test the predictive accuracy

of the model. The results demonstrated that the IEWS has more predictive accuracy than the other models used in the literature.

Although the statistical models are useful to managers and regulators with the authority to prevent the occurrence of failures (Ahn et al. 2000; Balcaen & Ooghe 2006), a number of drawbacks make them inapplicable as vital FEWS for business. They have various deficiencies such as: ignoring important sources of uncertainty in classification as an arbitrary definition of failure; data instability and arbitrary choice of optimization criteria; sampling selectivity; linearity assumption; and neglecting the time dimension of failure (Balcaen & Ooghe 2006). In addition, almost all existing statistical financial prediction models (Cole & Gunther 1995; Lane et al. 1986) have been criticized for their assumptions, which are more likely to be violated in the fields of finance and economics (Quek et al. 2009). Similarly, they are not able to identify the traits of financial distress that lead to bank failure and therefore function as black boxes (Tung et al. 2004). Conversely, the growing development of computational intelligence techniques has led researchers to employ new methods in FEWS. Since the main focus of this research is on the intelligent techniques, a short review of statistical methods and more detailed explanation on intelligent methods are provided.

2.4 Financial Failure Prediction: Intelligent Methods

The growing development and application of artificial intelligence and machine learning techniques, called ‘intelligent methods’, has led researchers to employ these methods in FEWS. Although they are time consuming and complex in some cases, these methods produce more accurate predictions than statistical methods. These models have demonstrated better performance than statistical methods and the idea of integrating them to achieve desirable results has been introduced in recent years, which has led to the appearance of hybrid models. These models may embed different techniques in an integrated framework or may use different methods separately, considering a weight for each one to generate a prediction. The result of these

methods in researches has shown that they usually outperform other sole intelligent techniques (Ahn et al. 2000). Ravi Kumar and Ravi (2007) provide a detailed review of the intelligent methods in the domain of bankruptcy prediction and demonstrate that their performance is better than statistical methods. All intelligent methods which have been applied for bank failure prediction can be categorized as illustrated in the following sections.

2.4.1 CASE-BASED REASONING

The approach which is newly introduced to this field of study is Case-Based Reasoning (CBR) which has significantly shown wonderful capability in classifying and predicting business failures (Li & Sun 2008, 2009a, 2009b, 2009c, 2010; Park & Han 2002). CBR is similar to human reasoning in the way it solves problems. It comprises four steps of: retrieving most similar cases; reusing the selected cases to solve the problem; revising the proposed solution to adapt to underlying problems; and retaining the obtained solution as a part of the new case. Even though many researchers were inspired to use CBR for business failure prediction, there is no special study reported for bank failure prediction. It is preferred to provide a quick review of some important CBR studies in business failure prediction. Park and Han (2002) integrated KNN and Analytic Hierarchy Process (AHP) to weight features. Next, they applied CBR to index and retrieve similar cases. The experimental results, which again were based on financial and non-financial data, concluded that the proposed approach outperforms other methods, including Logit and MDA. Li and Sun (2010) proposed an integrated model of CBR with the ELECTRE method. They used the ELECTRE method to define different similarity measures and consequently different CBR models. Next, they integrated these models to achieve the final output. They compared the proposed model with Logit, MDA and MLP and showed that it outperformed the others in terms of accuracy.

2.4.2 DECISION TREE

Decision Tree (DT), which is a powerful classifier technique, has also been used in bankruptcy prediction (Frydman et al. 1985). It uses a recursive partitioning algorithm to form rules on given data. For bank failure prediction, a binary DT is applied to classify banks through a set of IF-THEN rules on financial ratios. Marais et al. (1984) and Frydman et al. (1985) are two early studies which used decision tree for bank failure prediction. Very few papers have focused on DTs in Bank failure prediction since these early publications, despite the existence of many different DT algorithms including ID3, C4.5, C5 and CART (Gepp et al. 2010). Joos et al. (1998) used LA and C5 to predict credit classification for one of Belgium's largest banks. Nine models were created: three models from each technique based on three different data sets comprising a full set of financial variables, a reduced smaller set of financial variables and a set of qualitative variables. This study did not indicate that there was any obvious overall superiority between C5 and LA. Huarng et al. (2005) also used the C5 package, as well as being the first to apply CART to failure prediction. CART was found to be empirically superior to C5. Despite the lack of papers focused on DTs in bank failure prediction, various studies have used them as a comparison technique in bankruptcy prediction (Gepp et al. 2010).

2.4.3 SUPPORT VECTOR MACHINES

Support Vector Machine (SVM) which is an accurate machine learning technique for prediction which has been utilized in bankruptcy researches since 2000 (Ding et al. 2008; Hua et al. 2007; Min & Lee 2005; Shin et al. 2005). SVM uses a linear model and the optimal separating hyperplane to achieve the maximum separation between classes. The data points which are closest to the maximum margin hyperplane are called support vectors. Min and Lee (2005) proposed SVM for bankruptcy prediction. They proposed a grid-search technique using fivefold cross validation to discover the optimal parameter values of the kernel function of SVM. They compared the SVM with MDA, LOGIT and BPNN and used two kernels for SVM: RBF kernel and

polynomial kernel. According to experimental results, the classification rate of SVM is higher than that of other methods in both training and testing data sets.

2.4.4 FUZZY RULE-BASED CLASSIFIER

Fuzzy system has recently been introduced in this area to tackle the imprecise nature of financial forecasting (Alam et al. 2000; Tang & Chi 2005; Vigier & Terceño 2008). Fuzzy set theory provides a mathematical framework in which vague and conceptual phenomena can be rigorously studied. It is used to handle the vagueness of financial values and derive a set of IF-THEN fuzzy rules from given data to solve classification and prediction problems. Spanos et al. (1999) proposed a fuzzy rule generator method for bankruptcy prediction and compared it with LDA, LOGIT and PROBIT analysis. The proposed that the Fuzzy Rule Based Classifier (FRBC) would achieve greater accuracy than other methods and they concluded that the FRBC outperformed other methods for failure prediction on data gained from the Greek financial industry. Alam et al. (2000) used fuzzy clustering and two SOM-NNs to identify potentially failing banks. The results showed that both the fuzzy clustering and SOM-NNs showed promise in identifying potentially failing banks. Kumar and Ravi (2006) proposed a FRBC on the US banks bankruptcy data set. The task of classifier design was formulated as a multi-objective combinatorial optimization problem that aimed to maximize classification accuracy and minimize the number of rules. The results concluded that the proposed FRBC outperforms MLP neural network if two or four partitions were considered for the input data. Ravi et al. (2008) developed a FRBC by integrating NN, fuzzy classifier and modified threshold accepting algorithm for bank failure prediction on US banks data with financial rations under the CAMELS ranking system. They performed a comprehensive evaluation by comparing the proposed model with different types on NNs, SVM and CART. The results concluded that the proposed model outperforms the other models if two partitions are considered for the input data.

2.4.5 NEURAL NETWORK

Among intelligence techniques, Neural Networks (NN) are the most widely used in the domain of bank failure forecasting (Demyanyk & Hasan 2010). NN, which is based on the simulation of the biological neural network of the human neuron system, considers an interrelated group of artificial neurons and processes data through the connections among these neurons. The structure of the NN varies based upon data flow during the learning phase. It establishes a nonlinear and complex relationship among the input and output variables which is used to classify and predict the input data. The research conducted by Tam (1991) was one of the earliest studies to apply NN for bank failure prediction. Tam employed the Back Propagation Neural Network (BPNN) for bank failure. Data were obtained from Texas banks, one year and two years prior to failure. He selected the variables based on the CAMEL ranking system of the FDIC. The results showed that BPNN gains better predictive accuracy than other methods viz., DA, factor-logistic, K-Nearest Neighbor (K-NN) and ID3. Additionally, Tam and Kiang (1992) followed the research by comparing the performance of LDA, logistic regression, K-NN, ID3, Feed Forward Neural Network (FF_NN) and BPNN on bank failure prediction. They suggested that BPNN outperformed other techniques for a one-year prior training sample, whereas for a two-year prior training sample, DA outperformed others. However, BPNN outperformed others in both the one-year prior and the two-year prior testing samples. Bell (1997) compared logistic regression and BPNN in predicting bank failures. In this study, he used 28 candidate predictor variables. The architecture of BPNN was 12 input nodes, six hidden nodes and one output node. He concluded that neither the Logit nor the BPNN model dominated the other in terms of predictive ability, but for complex decision processes, BPNN was found to be better. Piramuthu et al. (1998) proposed a method called feature construction and used it with BPNN for bank failure prediction. The experiments performed were based on the Belgian bank data of 182 banks and the results concluded that BPNN with FC outperformed the plain BPNN in all data sets. Swicegood and Clark (2001) compared DA, BPNN and human judgment

in predicting bank failures. The results showed that that BPNN outperformed the other two models in identifying the status of underperformance banks. Lee et al. (2005) compared BPNN with Self-Organizing Feature Map Neural Network (SOM-NN), DA and logistic regressions. The data sample consisted of 168 banks taken from the Security and Exchange Commission (SEC) in an on-line database of the Korea Investors Service (KIS) Inc[3]. Fourfold cross-validation testing was used for all the models and the results concluded that the BPNN outperformed all other techniques. Boyacioglu et al. (2009) compared various types of NNs, Support Vector Machine (SVM) and multivariate statistical methods for the bank failure prediction problem in Turkey. They used similar financial ratios to those used in the CAMELS rating system. In the category of NN, different architectures are employed, namely Multilayer Perceptron (MLP), Self-Organizing Map (SOM) and Learning Vector Quantization (LVQ). The multivariate statistical methods tested are MDA, K-means cluster analysis, and Logit regression analysis. According to the comparison, MLP and LVQ can be considered the most successful models in predicting the financial failure of banks in the sample. Ravi and Pramodh (2008) proposed a Principal Component Neural Network (PCNN) architecture for bankruptcy prediction in commercial banks. In this architecture, the hidden layer is completely replaced by what is referred to as a 'principal component layer'. This layer consists of a few selected components that perform the function of hidden nodes. The authors tested the framework on data from Spanish and Turkish banks. According to the estimated results, hybrid models that combine PCNN and several other models to predict banking bankruptcy outperform other classifiers used in the literature.

2.4.6 Ensemble-based and Hybrid Methods

Although intelligent methods have produced better performance than statistical methods, the idea of integrating them to reach a desirable result has been introduced recent years and has led to the appearance of novel models. These may use different

[3] http://www.kisrating.com/eng/

methods separately, considering a weight for each one to produce prediction (Ensemble-based) or may embed different techniques in an integrated framework (Hybrid). The result of these methods in researches has shown that they usually outperform other stand-alone intelligent techniques (Ahn et al. 2000; Bahrammirzaee 2010; Verikas et al. 2010). Numerous ensemble-based models have been introduced in the bankruptcy prediction field in recent years; however, the most important ones are reviewed here. Hybrid models have also been widely applied for financial failure prediction in recent years.

In ensemble-based models, different aggregation strategies have been applied to aggregate the predictors: majority voting; averaging; and weighted averaging, each of which significantly affects predictive accuracy. Chan et al. (2006) aggregated a number of Radial Basis Function Neural Networks (RBFN) to form an ensemble model for failure prediction. To promote diversity of ensemble members, they performed bagging and selected features separately for each network trained on a separate bagged data set. Nominated features were those maximizing the mutual information between features and the class labels. The ensemble members were integrated using averaging, weighted averaging, and majority voting approaches to build three different ensemble models. When tested experimentally, ensembles achieved approximately the same performance. Yeung et al. (2007a, 2007b) also designed an ensemble of RBFNs to predict bankruptcy. With the aim of evolving varied ensemble members, diversity was conducted during the GA-based feature selection process by including a diversity term in the fitness function. Alfaro et al. (2008) as well as Cortes et al. (2007) applied an ensemble of DTs using the AdaBoost algorithm (Freund & Schapire 1995). AdaBoost gradually increases the number of ensemble members during the training process which is gradually more and more focused on misclassified training data points to boost the predictive accuracy. As a result, the ensemble model is formed by a linear combination of outputs of single predictors. When applying the AdaBoost ensemble of DTs to the bankruptcy data, the authors demonstrated a significant reduction in the test set error rate in comparison

with the error rate obtained from a single MLP. Tsai and Wu (2008) obtained unexpected bankruptcy prediction results from an ensemble of MLPs. The ensemble members were diversified through training data set manipulation and aggregated by a majority voting approach. On average, single MLP showed a higher accuracy than the ensemble. This is probably due to the very small data sets used to train the ensemble members as well as to the procedure applied to design the ensemble. Hua et al. (2007) suggested a simple combination of SVM and LR for bankruptcy prediction. If the output of the SVM is supported by LR with a large enough probability, the SVM decision is accepted. Otherwise, the decision may be modified depending on the interval the SVM output depends on. The empirical results showed superior performance of the ensemble model compared to the single SVM. Ravikumar and Ravi (2006) experimented with ensembles formed by various numbers of predictors. A set of seven classifiers was available: Adaptive Neuro Fuzzy Inference System (ANFIS), SVM, four types of RBFNs, and MLP. The majority voting rule has been used to aggregate the ensemble members. According to the empirical results, there was no superior ensemble model to report and the optimal size and structure of the ensemble were data dependent. Ravi et al. (2008) performed a comprehensive experiment by aggregating nine different classifiers to build an ensemble model for financial failure prediction. MLP, RBF, PNN, SVM, CART, FRBC, PCA-MLP, PCA-RBF, and PCA-PNN are the predictors used to build the ensemble. Majority voting and weighted averaging rules were used for the aggregation. Both ensembles outperformed the best single member, which was PCA-PNN.

Genetic Algorithm (GA), which mimics the principles of natural evolution to find the optimum solution of highly non-linear and non-convex problems, is usually used to find the optimum hyper-parameter value of a predictor. GA has been widely applied with other methods to form hybrid methods for failure prediction. A number of studies have used GA to find the optimum learning and topology parameters in MLP (Verikas et al. 2010). The results show that the GA-MLP methods outperform the MLP method in all data sets. Additionally, GA was used to select the parameters of

SVM and design SVM-based techniques for failure prediction (Ahn et al. 2006; Chen & Hsiao 2008; Min & Jeong 2009; Min et al. 2006). The results confirm the efficiency and superior performance of GA-SVM methods. Zhou and Tian (2007) suggested combining Rough Sets (RS) and SVM with wavelet kernel function. They applied RS to select the input features for the wavelet SVM model. They showed that the proposed hybrid model outperforms the SVM in bankruptcy prediction. Ahn et al. (2000) combined RS and MLP to increase the accuracy of bankruptcy prediction. They applied RS for feature selection and rule generation. Huysmans et al. (2006) combined MLP and SOM, aiming to exploit the good data exploration properties of SOM. MLP was trained first using financial input data. The input data used to train SOM consisted of the financial input data augmented with the output of the MLP. The results demonstrated the higher performance of the hybrid model in comparison with SOM and MLP. Lu et al. (2006) adopted the rule-based approach to achieve a transparent explanatory system for bankruptcy prediction. The authors extracted rules from a trained neural network, then applied the GA to obtain ultimate classification rules.

2.4.7 FUZZY NEURAL NETWORK

One of the most important advantages of NN is its adaptivity. NN can automatically adjust its connection weights using a learning algorithm to optimize its behaviour for applications such as pattern recognizers, system controller and predictors. Through a training process, NN can learn to estimate the input–output function without any mathematical model. However, analysis of a trained network is difficult, as the network itself is treated as a black box. It is difficult to interpret and relate the connection weights to the dynamics of the problem domain. Thus, a NN lacks explanatory capabilities for its outcome, which is certainly undesirable. Furthermore, it is difficult to decide the number of nodes and hidden layers for a particular application (Quek et al. 2009).

On the other hand, the strength of fuzzy logic lies in its capabilities in modeling vagueness, handling uncertainty and supporting human-type inference skills. The behaviour of this fuzzy-logic-based system is governed by fuzzy IF–THEN rules expressed in the form IF A THEN B, where A and B are fuzzy sets. Thus, a fuzzy system is described in natural or synthetic language that is comprehensible by humans. Unlike a conventional set, where an element either belongs or does not belong to a set, a fuzzy set expresses the degree to which an element belongs to the set. The advantage of a fuzzy system lies in its ability to perform the inference mechanism under cognitive uncertainty. Its mechanism can be explained on the basis of fuzzy rules and its performance can be adjusted by tuning the rules. A fuzzy system will be appropriate if sufficient expert knowledge about the problem is available to design and tune the membership functions and the underlying rule base. The process of tuning the membership functions and rules is a trial-and-error process (Chen and Quek, 2003) and is one of the main drawbacks of the fuzzy system, which as a result, restricts the fuzzy system to areas where expert knowledge is available (Quek et al. 2009).

Integrating NN and fuzzy system gives a hybrid model named Fuzzy Neural Network (FNN) that has the merits of both methods. In FNN, NN can be employed to automate the process of tuning membership functions and deriving IF–THEN rules. At the same time, the connectionist hybrid structure becomes transparent and the results or outputs of FNN become explainable. All types of FNN can be generally categorized into two groups: (1) FNNs with self-tuning ability which requires an initial rule base to be specified before training (Berenji & Khedkar 1992; Jang 1993). (2) FNNs which have the capability to automatically create fuzzy rules from numerical training data (Lin & Lin 1997; Quek & Zhou 1999; Zhou & Quek 1996). The main advantage of the latter category is that it can extract knowledge from implicit patterns in numerical data by automatically generating a fuzzy rule base. Moreover, it does not need to have prior knowledge, such as the number of clusters (fuzzy sets) for each variable and characteristics of these clusters. FNN is robust, fault tolerant and capable of acquiring

new knowledge and performing a human-like inference mechanism under cognitive uncertainty. It has been applied as a robust hybrid classifier and forecaster tool in different fields (Ang et al. 2003; Jang 1993; Khotanzad et al. 2000; Lin & Lin 1997; Lin & Lee 1996; Nauck et al. 1997; Pasquier et al. 2001; Peymanfar et al. 2007; Quek & Zhou 1999; Sim et al. 2006b; Singh et al. 2008; Tung & Quek 2002, 2004; Zhang & Morris 1999; Zhou & Quek 1996). In recent research, different kinds of FNNs are also used to classify and predict financial failures (Chen et al. 2009b; Lin et al. 2008; Ng et al. 2008; Quek et al. 2009; Tung et al. 2004). By combining fuzzy sets theory and the MLP, Gorzalczany and Piasta (1999) designed a FNN classifier for bankruptcy prediction. The fuzzy sets-based input module allows the input of purely numerical data as well as qualitative, linguistic data that may be used to characterize the decision-making process. The authors demonstrated the superiority of the FNN classifier over the rough sets-based technique, C4.5 decision tree, and the rule induction system CN2 (Clark & Niblett 1989). Tung et al. (2004) proposed the Generic Self-organizing Fuzzy Neural Network (GenSoFNN) to predict bank failure. The proposed FNN also consists of five layers: input layer, antecedent matching layer, rule-based layer, consequent derivation layer, and output layer. Parameters of the network are learned through the gradient descent. The base of IF-THEN rules designed during training provides insight into the contribution of the selected financial covariates to the bank failure. Thus, it is possible to analyze the reasons behind the bankruptcy and identify the symptoms of financial distress. Tung et al. (2004) selected nine significant financial variables, and performed experiments using 21-year historical data of USA banks between 1980 and 2000. Even though the GenSoFNN has been found to be more successful than Cox‘s Proportional Hazards model, a slightly lower prediction accuracy is obtained from the GenSoFNN compared to MLP. The authors advocate using the GenSoFNN network due its transparency. Lee et al. (2006) studied the efficiency of several training techniques applied to the POPFNN-CRI(S) (Ang et al. 2003), which was then used to predict bank failure. As is often the case in FNNs, the network consisted of five layers: input,

antecedent, rule base, consequence, and output. They investigated the effect of missing data on bank failure prediction and found that it does not affect the outcome. Nguyen et al. (2008) aimed to construct a novel fuzzy neural Cerebellar Model Articulation Controller (CMAC) for bank failure prediction. They applied a nature inspiration motivated by the famous Chinese ancient Ying–Yang philosophy to find the optimal fuzzy sets, and a Truth Value Restriction (TVR) inference scheme to derive the truth-values of the rule weights. The proposed Ying–Yang FCMAC network can identify the inherent traits of financial distress based on financial features. The advantages of the proposed model are its fuzzification technique using Bayesian Ying–Yang learning and its TVR inference scheme. The experiments' design and data are similar to those conducted by Tung et al. (2004). Three sets of experiments were performed – bank failure classification based on the last available financial record and prediction using financial records one and two years prior to the last available financial statements. The performance of the proposed Ying–Yang FCMAC network was very encouraging, as it achieved 95% average accuracy. Oentaryo et al. (2008) combined the architecture of GenSoFNN-CRI (Tung et al. 2004) and the inference scheme of FCMAC-Yager (Sim et al. 2006a) to emulate the sequential learning paradigm of the hippocampus in the brain to synthesize low-level numerical data to high-level declarative fuzzy rules. The proposed GenSoFNN-Yager exhibits simple and conceptually firm computational steps that correspond closely to a plausible human logical reasoning and decision-making. The authors claimed that the proposed model outperforms its predecessors since it adopts an online learning mechanism that can identify the relevant rules based on a single sample, while other systems need to first view the entirety of the data before they can build the rule base due to their offline (batched) learning approach. The empirical results demonstrate its promising output and superior performance compared to K-NN, MLP, and GenSoFNN-CRI. The accuracy of the proposed model is slightly less than the FCMAC-Yager, but the number of rules it requires to forecast accurately is twenty times less than required by FCMAC-Yager. Ng et al. (2008) proposed a FNN, called

FCMAC-CRI, as a new approach to tackling the problem of bank failure prediction using localized learning. The study was inspired by the fact that localized learning is similar to neocortex semantic associative memory, which is superior to the hippocampal form of global learning. The FCMAC-CRI network is a FNN whose operations are defined by the fuzzy inference scheme, compositional rule of inference (CRI) and its interactive relations among the selected features are captured in the form of highly intuitive fuzzy IF–THEN rules, which form the knowledge base of FEWS. The performance of the FCMAC-CRI was benchmarked against that of the Cox's proportional hazard model and GenSoFNN-CRI, which is based on globalised training technique. The experiments, whose design and data were similar to the experiments performed by Tung et al. (2004) revealed that FCMAC-CRI consistently outperforms the Cox's model and GenSoFNN-CRI. The main advantage of the proposed model is its significant discrimination and interpretation ability over FNN which are based global learning. Chen et al. (2009b) presented a simple FNN for bankruptcy prediction. The paper did not aim to establish the best FNN for bankruptcy prediction problem; instead, it recommended FNN as an alternative method to solve the problem. The empirical results showed that FNN had a better accuracy rate, lower misclassification cost and higher detecting power than Logit regression. They concluded that because FNN provides a much more detailed relationship among the variables than the traditional statistical method and NN, it would be an appropriate turnkey system for detecting bankruptcy; both practitioners and academics would be expected to interact naturally with it. This 'exploratory relationship' as a lens in understanding bankruptcy patterns implies that new knowledge could be pursued and continuously updated.

The main advantages of FNNs are their consistent fuzzy rule base gained from fuzzy systems along with their learning ability and accuracy obtained from NN for the prevention of pending financial failures. These models not only predict the financial status of a corporate concern relatively accurately, but also provide a knowledge base which may be used to make decisions and prevent such circumstances before they

spiral out of control. However, they have drawbacks which are worthy of mention: Firstly, their predictive accuracy suffers in comparison with NN, which is one of the most accurate prediction method with the limitation of functioning as a 'black box' (Ng et al. 2008). Secondly, the success of the FNN depends on the choice of parameters and data features used in the training processes. Determining the optimal system parameters is presently a trial-and-error process in which the relevance of features to the problem in hand may not be known a priori. To resolve these problems, incorporation of the higher-level (meta-cognitive) mechanisms in the FNN model is highly desirable. Employing this architecture will allow a more comprehensive modeling of general human intelligence and accordingly a better intelligent framework for the future (Ng et al. 2008).

2.5 Class Imbalance Problem and Solutions

The financial distress prediction problem including bankruptcy and bank failure prediction is inherently categorized as a class imbalance problem. This problem occurs when the number of instances of one class is much lower than the instances of other classes. Bank failure prediction particularly focuses on a two class imbalance data-set problem, where there is only one positive class (failed corporate) with a lower number of examples, and one negative class (survived corporate) with a higher number of examples. The problem appears in bank failure prediction because very few banks go failure in proportion and also it is difficult to obtain all the relevant data from them.

The problem is extremely important, because it is pervasive in a large number of real-world applications such as medical diagnosis, fraud detection, network intrusion detection, modern manufacturing plants and financial failure prediction (Haibo & Garcia 2009; Sun et al. 2009). Over recent years, the imbalanced data set problem has demanded considerable attention in the field of classification and prediction (Chawla

et al. 2004; YANG 2006) and many theoretical and experimental studies have investigated the influence of the imbalance problem on predictors and classifiers. It has been suggested that the skewed data distribution is not the only parameter that describes the class imbalance problem. Other influential factors such as small sample size, class separability and within-class concepts are other parts of the nature of the class imbalance problem.

As a result of the comprehensive review of deficiencies of well-developed classifier and predictor algorithms such as DT, NN, Bayesian classification, SVM and Associative classification and K-NN when encountering the imbalanced data-set problem, it is generally concluded that during learning from an imbalanced data set, the classifier or predictor will obtain a high predictive accuracy for the majority class but will predict poorly for the minority class, which is equally necessary in prediction (Weiss 2004). Likewise, the classifier may consider the minority class as noise, which is then ignored and may result in a negative influence on the ability of most classification methods (Japkowicz & Stephen 2002; Orriols-Puig & Bernadó-Mansilla 2009).

Since this problem can significantly affect the performance of prediction model and reduce the accuracy, many studies have proposed techniques and methods to tackle this problem in various applications. These techniques can be categorized into four main groups: data level methods; algorithm level methods; cost-sensitive learning and boosting approaches.

2.5.1 DATA LEVEL METHODS

At the data level, the objective is to rebalance the class distribution by resampling the data space. Data level solutions include many different forms of resampling such as random oversampling the minority class with replacement, random undersampling the majority class, informatively oversampling the minority class, informatively undersampling the majority class, oversampling with generation of new synthetic samples, and combinations of the above techniques. The main advantage of data level

methods is that they are more independent and adaptable and can be used with most classifiers and predictors.

Random undersampling is a non-heuristic method that aims to balance class distribution through the random elimination of majority class examples. The major drawback of random undersampling is that this method can discard potentially useful data that could be important for the induction process. Condensed Nearest Neighbour rule (CNN) (Gowda & Krishna 1979) is an early undersampling method to find a consistent subset of examples. This procedure eliminates the examples from the majority class that are distant from the decision border, since these sorts of examples might be considered less relevant for learning. Tomek links (Tomek 1976), which is an undersampling method, can also be used as a data cleaning method. As an undersampling method, only examples belonging to the majority class are eliminated, and as a data cleaning method, examples of both classes are removed. One-Sided Selection (OSS) (Kubat & Matwin 1997) is an undersampling method resulting from the application of Tomek links followed by the application of CNN. Tomek links are used as an undersampling method and remove noisy and borderline majority class examples. CNN is applied to remove examples from the majority class that are distant from the decision border. The remaining samples, i.e., safe majority class examples and all minority class examples are used for learning. Wilson's Edited Nearest Neighbour rule (ENN) (Batista et al. 2004) removes majority class examples whose class label differs from the class of at least two of its three nearest neighbours.

Random oversampling is a non-heuristic method that aims to balance class distribution through the random replication of minority class examples. Several authors agree that random oversampling can increase the likelihood of overfitting occurring, since it makes exact copies of the minority class examples. Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al. 2002) is an oversampling method. Some researches (Fernández et al. 2009a, 2009b; Fernández et al. 2008) show that oversampling methods, particularly SMOTE, significantly improve the predictive accuracy of FRBC, which is the main focus of this study. This technique

enables the minority instances to be oversampled by producing synthetic examples along the line segments joining any/all of the K minority examples nearest neighbors. To do this, synthetic examples are created by multiplying the distance between the sample under consideration and its nearest neighbor by a random number between 0 and 1, and adding it to the considered sample. This approach effectively makes the decision region of minority class more general, and therefore the over-fitting problem is avoided and minority examples spread further into the majority samples.

SMOTE + Tomek links (Fernández et al. 2008) is a hybrid method that applies Tomek links to the oversampled training set of SMOTE as a data cleaning method. Thus, instead of removing only the majority class examples that form Tomek links, examples from both classes are removed. SMOTE+ENN (Fernández et al. 2008) applies ENN to remove examples from both classes, so it is expected that it will provide a more in-depth data cleaning.

2.5.2 ALGORITHM LEVEL METHODS

The methods at the algorithm level try to adapt and adjust the learning algorithms to reinforce learning with regards to the minority class. To develop an algorithmic solution, knowledge of both the corresponding classifier or predictor and the application domain is required, especially a thorough understanding of why the learning algorithm fails when the class distribution of available data is uneven (Sun et al. 2009). Generally, a common strategy to deal with the class imbalance problem is to choose an appropriate inductive bias. For instance, adjusting the probabilistic estimate at the tree leaf or developing new pruning techniques (Zadrozny & Elkan 2001) are methods to handle class imbalance problem in decision trees. For SVMs, using different penalty constants for different classes or adjusting the class boundary based on kernel-alignment ideal (Wu & Chang 2003) are reported solutions. In one-class (recognition-based) learning, the classifier or predictor is trained with only examples of the target class. This approach does not try to partition the hypothesis space with boundaries that separate positive and negative examples, but attempts to

make boundaries which surround the target concept. It measures the amount of similarity between a query object and the target class, where a threshold on the similarity value is introduced. NN and SVM have been studied in the context of the one-class learning approach (Japkowicz 2001; Raskutti & Kowalczyk 2004). The results demonstrate that under certain conditions, the one-class approach is superior to two-class learning. However, many machine learning algorithms, such as DT, do not function unless the training data includes examples from different classes.

2.5.3 Cost-sensitive Learning Methods

Cost-sensitive learning incorporates both the data and algorithm level methods. It assumes higher misclassification costs with samples in the minority class and tries to minimize the high cost errors (Ling et al. 2004). Cost-sensitive learning takes the varying costs of different misclassification types into account (Margineantu 2002). A cost matrix encodes the penalty of classifying samples from one class as another class. In dealing with the class imbalance problem, the recognition importance of positive instances is higher than that of negative instances. Hence, the cost of misclassifying a positive instance outweighs the cost of misclassifying a negative one. The cost-sensitive learning process then tries to minimize the number of high cost errors and the total misclassification cost using the cost matrix during training and generates a model that has the lowest cost.

2.5.4 Boosting Approaches

Boosting algorithms are iterative algorithms that allocate different weights on the training distribution in each iteration. Boosting increases the weights associated with the incorrectly classified examples and decreases the weights associated with the correctly classified examples during the training process. This forces the learning-based model to focus more on the incorrectly classified or predicted examples in the next iteration. Because minority instances are more error-prone than majority instances, it will increase the weights of the samples associated with the minority

class. They are also regarded as solutions at the data level since they update the data space by weighting samples. Several boosting algorithms are also reported as meta-techniques which are applicable to most classifier and predictor algorithms (Kotsiantis et al. 2006; Ting 2000). AdaCost (Fan et al. 1999) has been empirically shown to produce lower cumulative misclassification costs and has thus proved to be effective in addressing the class imbalance problem. Joshi et al. (2001) proposed Rare-Boost for class imbalance problem. It scales false-positive examples in proportion to how well they are distinguished from true-positive examples and scales false-positive examples in proportion to how well they are distinguished from true-negative examples. Another algorithm that uses boosting to handle the imbalanced data-set problem is SMOTEBoost proposed by Chawla et al. (2003). This algorithm recognizes that boosting may suffer from the same problem as overfitting, since boosting tends to weight samples belonging to the minority class more than those belonging to the majority class, effectively duplicating some of the samples belonging to the minority class. Hence, SMOTEBoost alters the distribution by adding new minority-class samples using the SMOTE algorithm, instead of changing the distribution of training data by updating the weights associated with each example.

2.5.5 Evaluation Measures

Evaluation measures play a crucial role in both assessing the model performance and guiding the model training. Traditionally, accuracy, indicated by Equation (2.1) for the bi-class scenario, is the most commonly used measure for these purposes.

$$\text{Accuracy: } Acc = \frac{TP+TN}{TP+FN+FP+TN} \tag{2.1}$$

where the components are computed using a confusion matrix as indicated in Table 2.3 . This addresses the number of correctly and incorrectly predicted samples for each class. TP (TN) contains the number of instances that are predicted correctly positive (negative). FP (FN) is the number of examples that are predicted wrongly positive (negative), which actually belong to the negative (positive) class.

TABLE 2.3: CONFUSION MATRIX

	Predicted as Positive	Predicted as Negative
Actually Positive	True Positive (TP)	False Negative (FN)
Actually Negative	False Positive (FP)	True Positive (TP)

For dealing with the class imbalance problem, accuracy is no longer a reliable measure since the minority class has very little impact on accuracy compared to the majority class. For example, in a problem where a minority class is represented by only 1% of the training data, a simple strategy can be one that predicts the majority class label for every sample. It can achieve a high accuracy of 99%. However, this measurement is meaningless to some applications where the learning concern is the prediction label of instances in the minority class.

Given a data set with imbalanced class distribution, the classifier or predictor performance on the minority class is usually unsatisfactory. To remedy this, the learning objective can be: (1) balancing the identification abilities between the two classes; and/or (2) improving the recognition success on the minority class. With respect to different learning objectives, the performance of the model should be evaluated by different measures (Sun et al. 2009). For different evaluation criteria, several measures are derived from a confusion matrix as follows:

True Positive Rate (Sensitivity): $TP_{rate} = \frac{TP}{TP+FN}$ (2.2)

True Negative Rate (Specificity): $TN_{rate} = \frac{TN}{FP+TN}$ (2.3)

False Positive Rate: $FP_{rate} = \frac{FP}{TN+FP}$ (2.4)

False Negative Rate: $FN_{rate} = \frac{FN}{TP+FN}$ (2.5)

Positive Predictive Value: $PP_{value} = \frac{TP}{TP+FP}$ (2.6)

Negative Predictive Value: $NP_{rate} = \frac{TN}{TN+FN}$ (2.7)

F-measure: $F = \frac{2 \times TP_{rate} \times PP_{value}}{TP_{rate} + PP_{value}}$ (2.8)

G-Mean: $GM = \sqrt{TP_{rate} \times TN_{rate}}$ (2.9)

If only the performance of the positive class is considered, two measures are important: True Positive Rate (TP_{rate}) and Positive Predictive Value (PP_{value}). In information retrieval, TP_{rate} is defined as recall (R) denoting the percentage of retrieved objects that are relevant. PP_{value} is defined as precision (P) denoting the percentage of relevant objects that are detected for retrieval. F-measure (F), which was proposed by Lewis and Gale (1994) to integrate these two measures as an average, is a harmonic mean between recall and precision. The harmonic mean of two numbers tends to be closer to the smaller of the two. Hence, a high F-measure value ensures that both recall and precision are reasonably high. G-Mean (GM), which is a geometric mean of sensitivity and specificity, is applied when performances of both classes are expected to be high simultaneously (Barandela et al. 2003; Kubat et al. 1998). GM measures the balanced performance of a learning algorithm between these two classes.

Some models, such as Bayesian Network inference or some NNs, assign a probabilistic score to their prediction. Class prediction can be changed by varying the threshold. Each threshold value generates a pair of measurements of (FP_{rate}, TP_{rate}). The Receiver Operating Characteristics (ROC) graph, which is showed in Figure 2.3 as an example, is created by plotting these pairs. The ideal model is one that obtains 1 for TP_{rate} and 0 for FP_{rate}.Therefore; a good model should be located as close as possible to the upper left corner of the diagram. However, when comparing models, it is hard to claim a winner unless one curve clearly dominates the others over the entire space. For instance, according to the graph in Figure 2.3, it is difficult to distinguish model A or B as the superior model. The Area Under a ROC Curve (AUC) provides a single measure of performance to judge which model is better on average. A model with higher AUC can be considered as superior model.

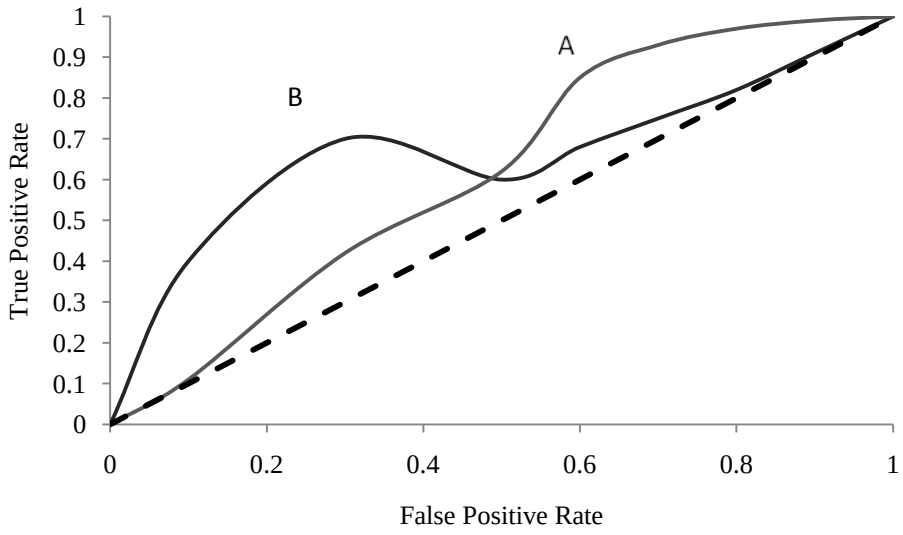


FIGURE 2.3: ROC CURVES FOR TWO DIFFERENT MODELS

2.6 TRANSFER LEARNING

Although machine learning technologies have attracted a remarkable level of attention in researches in different computational fields, including prediction, most of them work under the common assumption that the training data (source domain) and the test data (target domain) have identical feature spaces with underlying distribution. As a result, once the feature space or the feature distribution of the test data changes, the prediction models cannot be used and must be rebuilt and retrained from scratch using newly-collected training data, which is very expensive, if not practically impossible (Pan & Yang 2010). Similarly, since learning-based models need adequate labeled data for training, it is nearly impossible to establish a learning-based model for a domain (target domain) which has very few labeled data available for supervised learning. If we can transfer and exploit the knowledge from an existing similar but not identical domain (source domain) with plenty of labeled data, however, we can pave the way for construction of the learning-based model for the target domain. In real world scenarios, particularly in the finance industry, there are many situations in

which very few labeled data are available, and collecting new labeled training data and forming a particular model are practically impossible. For instance, there are plenty of labeled data available to construct a prediction model to specify bank status in the state of California (source domain), whereas there are very few samples available for the banking system in the state of Texas (target domain). Since they might not have identical feature spaces, it is not possible to use the same model for both domains. However, they are similar and have common features, which may assist in the employment of the prediction model in the target domain. To transfer the knowledge between these two domains, we can explore the similarities, construct the cross-domain relationship between the two domains with different but related feature spaces, and bridge the gap between two domains through this relationship.

Transfer learning has emerged in the computer science literature as a means of transferring knowledge from a source domain to a target domain. Unlike traditional machine learning and semi-supervised algorithms (Blum & Mitchell 1998; Joachims 1999b; Nigam et al. 2000; Zhu 2005), transfer learning considers that the domains of the training data and the test data may be different (Fung et al. 2006). Traditional machine learning algorithms make predictions on the future data using mathematical models that are trained on previously collected labeled or unlabeled training data which is the same as future data (Baralis et al. 2008; Kuncheva & Rodriguez 2007; Yin et al. 2006). Transfer learning, in contrast, allows the domains, tasks, and distributions used in training and testing to be different. In the real world, we observe many examples of transfer learning. For example, we may find that learning to recognize apples might help to recognize pears. Similarly, learning to play the electronic organ may facilitate learning the piano. The study of transfer learning has been inspired by the fact that human beings can utilize previously-acquired knowledge to solve new but similar problems much more quickly and effectively. The fundamental motivation for transfer learning in the field of machine learning focuses on the need for lifelong machine learning methods that retain and reuse previously learned knowledge. Research on transfer learning has been undertaken

since 1995 under a variety of names: learning to learn; life-long learning; knowledge transfer; meta learning; inductive transfer; knowledge consolidation; context sensitive learning and multi-task learning (Pan & Yang 2010). In 2005, the Broad Agency Announcement (BAA) of the Defense Advanced Research Projects Agency (DARPA)'s Information Processing Technology Office (IPTO) [4] gave a new mission to transfer learning: the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks. In this definition, transfer learning aims to extract the knowledge from one or more source tasks and to then apply the knowledge to a target task. Figures 2.4 and Figures 2.5 shows the difference between the learning processes of traditional and transfer learning techniques (Pan & Yang 2010). As can be seen, traditional machine learning techniques try to learn each task from scratch, while transfer learning techniques try to transfer the knowledge from some tasks and/or domains to a target task when the latter has fewer high-quality training data.

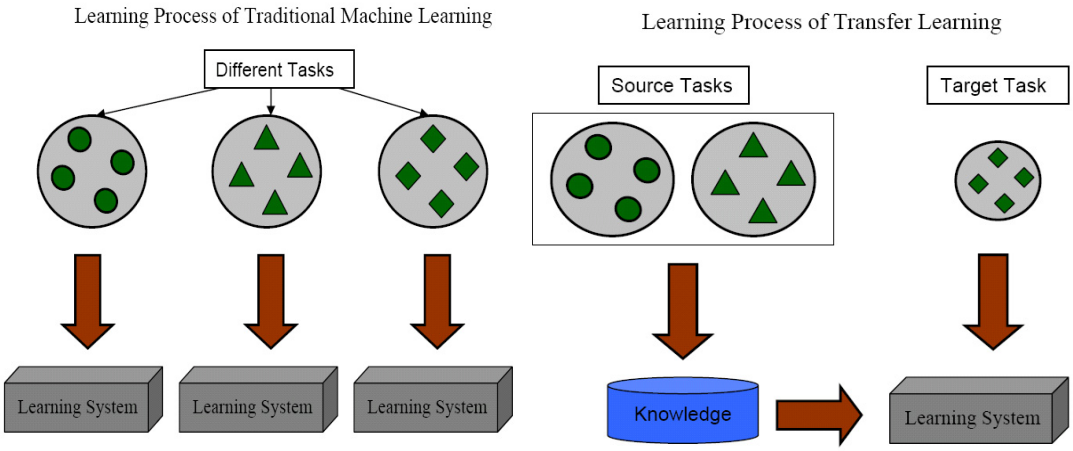


FIGURE 2.4: DIFFERENT LEARNING PROCESSES BETWEEN TRADITIONAL MACHINE LEARNING AND TRANSFER LEARNING

[4] http://www.darpa.mil/ipto/programs/tl/tl.asp

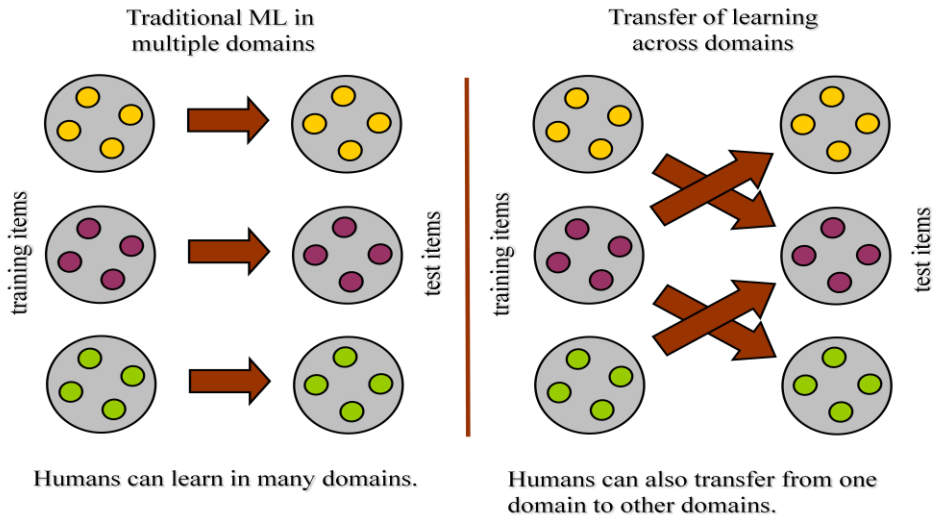


FIGURE 2.5: DIFFERENT LEARNING PROCESSES BETWEEN TRADITIONAL MACHINE LEARNING AND TRANSFER LEARNING

It must be emphasized that even though transfer learning and concept drift (Moreno-Torres et al. 2012) are studied as the same technique in many database and data mining researches, they are very different. One of the major differences is that the entire training and test data are available to learn in transductive transfer learning while there is only a small number of test data for learning in concept drift (Yang 2009). Additionally, the semi-supervised learning (Chawla & Karakoulas 2005) is a new technique which also exploits the unlabeled data in training the prediction/classification model. In contrary to the transfer learning, it assumes that the training and test data drive from the same domain and have the same distribution.

2.6.1 DEFINITIONS AND NOTATIONS

The notations and definitions that will be used throughout the section are introduced. According to the definitions, we then categorize the various settings of transfer learning algorithms that exist in the literature of machine learning.

Definition 2.1 (Domain)(Pan & Yang 2010) A domain, which is denoted by $D = \{\chi, P(X)\}$, consists of two components:

(1) Feature space χ ; and

(2) Marginal probability distribution $P(X)$, where $X = \{x_1, \dots, x_n\} \in \chi$.

For example, if the learning task is bank failure prediction**,** χ is the financial ratios that are applied for prediction, X is the set of all instances (banks) and $P(X)$ is the marginal distribution of these instances. In general, if two domains are different, they may have different feature spaces or different marginal probability distributions.

Definition 2.2 (Task) (Pan & Yang 2010) A task, which is denoted by $T = \{Y, f(\cdot)\}$, consists of two components:

(1) A label space $Y = \{y_1, \dots, y_m\}$; and

(2) An objective predictive function$f(\cdot)$ which is not observed and to be learned by pairs $\{x_i, y_i\}$.

The function $f(\cdot)$ can be used to predict the corresponding label, $f(x_i)$, of a new instance x_i. From a probabilistic viewpoint, $f(x_i)$ can be written as $P(y_i|x_i)$. In the bank failure prediction example, which is a binary prediction task, y_i can be the label of failed or survived. More specifically, the source domain can be donated as $D_s = \{(x_{s_1}, y_{s_1}), \dots, (x_{s_n}, y_{s_n})\}$ where $x_{s_i} \in \chi_s$ is the source instance or bank in bank failure prediction example and $y_{s_i} \in Y_s$ is the corresponding class label which can be failed or survived for bank failure prediction. Similarly, the target domain can be denoted as $D_t = \{(x_{t_1}, y_{t_1}), \dots, (x_{t_n}, y_{t_n})\}$ where $x_t \in \chi_t$ is the target instance and $y_{t_i} \in Y_t$ is the corresponding class label and in most scenarios $t_n \ll s_n$.

Definition 2.3 (Transfer learning) (Pan & Yang 2010) Given a source domain D_s and learning task T_s, a target domain D_t and learning task T_t, transfer learning aims to improve the learning of the target predictive function $f_t(\cdot)$ in D_t using the knowledge in D_s and T_s where $D_s \neq D_t$ or $T_s \neq T_t$.

In the above definition, the condition $D_s \neq D_t$ implies that either $\chi_s \neq \chi_t$ or $P_s(X) \neq P_t(X)$. For example, in the bank failure prediction example, this means that between a source banking system and a target banking system, either the financial features are different between the two domains, or the marginal distributions of banks

are different. Similarly, the condition $T_s \neq T_t$ implies that either $Y_s \neq Y_t$ or $f_s(\cdot) \neq f_t(\cdot)$. For instance, it corresponds to a situation in which the source banking system has binary class labels of failed and survived, whereas the target banking system has more than two class labels, or the source prediction model and target prediction model classify an identical bank in different class labels. In addition, there are some explicit or implicit relationships among feature spaces off two domains that we imply that the source domain and target domain are related. It should be mentioned that when target and source domain are the same ($D_s = D_t$) and their learning tasks are also the same ($T_s = T_t$), the learning problem becomes a traditional machine learning problem.

2.6.2 TRANSDUCTIVE TRANSFER LEARNING

According to the uniform definition of transfer learning introduced by Definition 2.3, the transfer learning techniques can be divided into three main categories (Pan & Yang 2010): (1) Inductive transfer learning, in which the learning task in the target domain is different from the target task in the source domain ($T_s \neq T_t$); (2) Unsupervised transfer, learning which is similar to the inductive transfer learning but focuses on solving unsupervised learning tasks in the target domain such as clustering, dimensionality reduction and density estimation ($T_s \neq T_t$); and (3) Transductive transfer learning (domain adaptation), in which the learning tasks are the same in both domains, while the source and target domains are different ($T_s = T_t,\ D_s \neq D_t$). In the literature, transductive transfer learning, domain adaptation, covariate shift, sample selection bias, transfer learning, multi-task learning, robust learning, and concept drift are all terms which have been used to handle the related scenarios. More specifically, when the method aims to optimize the performance on multiple tasks or domains simultaneously, it is considered as multi-task learning. If it optimizes performance on one domain, given training data that is from a different but related domain, it is considered as transductive transfer learning or domain adaptation. Transfer learning and transductive transfer learning have been often used interchangeably with domain

adaptation. Concept drift refers to a scenario in which data arrives sequentially with changing distribution, and the goal is to predict the next batch given the previously-arrived data (Klinkenberg & Joachims 2000).The goal of robust learning is to build a classifier that is less sensitive to certain types of changes, such as feature change or deletion in the test data. In addition, unsupervised domain adaptation can be considered as a form of semi-supervised learning, but it assumes that the labeled training data and the unlabeled test data are drawn from different distributions.

Since the focus of this study is on the last category of transfer learning techniques, called domain adaptation, this category is reflected and the related studies are reviewed. Domain adaptation can be defined more specifically as follows:

Definition 2.4 (Domain Adaptation) (Pan & Yang 2010) A category of transfer learning in which $T_s = T_t$ and $D_s \neq D_t$ which implies that either $\chi_t \neq \chi_s$ or $P_t(X) \neq P_s(X)$.

A distinction exists between supervised domain adaptation, which assumes some labeled data in the target domain, vs. unsupervised domain adaptation, which assumes only labeled data from the source domain and unlabeled data from the target domain. In this situation, no labeled data in the target domain are available while a large quantity of labeled data in the source domain is available. In addition, according to the above definition, domain adaptation can also be divided into two cases: (1) The feature spaces between domains are the same ($\chi_t = \chi_s$), but the marginal probability distributions of the input data are different ($P_t(X) \neq P_s(X)$). (2) The feature spaces between the source and target domains are different ($\chi_t \neq \chi_s$).

The existing techniques and methods, which have been used to handle the domain adaptation problem, can be divided into four main classes (Margolis 2011): (1) Instance weighting for covariate shift methods which weight samples in the source domain to match the target domain; (2) Self-labeling methods which include unlabeled target domain samples into the training process and initialize their labels and then iteratively refining the labels; (3) Feature representation methods which try

to find a new feature representation of the data, either to make the target and source distributions look similar, or to find an abstracted representation for domain-specific features; and (4) Cluster-based learning methods rely on the assumption that samples connected by high-density paths are likely to have the same label.

2.6.2.1 Instance Weighting for Covariate Shift Methods

From the perspective of domain adaptation for machine learning, the covariate shift assumption implies that the data distribution differs ($P_s(X) \neq P_t(X)$), but the conditional label probabilities are the same ($P_s(Y|X) \equiv P_{s\cup t}(Y|X) \equiv P_t(Y|X)$). Also, $P_t(X)$ is assumed to have support within that of $P_s(X)$ (Huang et al. 2007; Quionero-Candela et al. 2009). The covariate shift scenario might arise in cases where the training data has been biased toward one region of the input space or is selected in a non-I.I.D. manner. It is closely related to the idea of sample-selection bias which has long been studied in statistics (Heckman 1977) and in recent years it has been explored for machine-learning. Huang et al. (2007) proposed a novel procedure called Kernel Mean Matching (KMM) to estimate weights on each instance in the source domain, based on the goal of making the weighted distribution of the source domain look similar to the distribution of the target domain. Sugiyama et al. (2008) and Tsuboi et al. (2009) proposed a similar idea called the Kullback-Leibler Importance Estimation Procedure (KLIEP). Here too the goal is to estimate weights to maximize similarity between the target and weight-corrected source distributions. Zadrozny (2004) adopted a generative model for covariate shift in which a binary random variable determines whether or not a target domain sample is selected into the training set of the source domain. Cortes et al. (2008) proposed a weight estimation method based on clustering all the data and estimating one weight value for the training samples in each cluster, based on the proportion of source samples. Ren et al.(2008) proposed a different cluster-based method which selects training examples to balance the distribution across clusters. Rosset et al. (2005) proposed a method-of-moments procedure for estimating the sampling distribution: they assumed a

parametric form for the distribution, and solved the parameters by equating empirical moments of features in the training set with weighted empirical moments in the whole data set.

2.6.2.2 SELF-LABELING METHODS

Self-labeling methods, which include self-training and co-training approaches, train an initial model based on the labeled source data iteratively. They use the initial model to estimate the labels on the target data, and then use the estimated labels to build another model.

In self-training, source domain data are used to train an initial model, which is then used to estimate labels for target domain instances. In the next round, target domain data are incorporated to train a new model. This is carried out repeatedly, either for a fixed number of rounds, or until convergence. There are various approaches to how to select the target domain instances. Some approaches add only the top k samples with the highest label confidence on each round, while others use all the data on each round, repeatedly adjusting the labels for those data on subsequent rounds. Self-training has a close relationship with the Expectation Maximization (EM) algorithm, which has a hard and soft version. The hard version adds samples with single certain labels while the soft version assigns label confidences when fitting the model. These methods have been applied in both supervised and unsupervised domain adaptation settings. EM does not deal explicitly with the fact that $P_s(X) \neq P_t(X)$, but since it attempts to model both source and target domains simultaneously, it will do a better job of generalizing to the target domain compared to only modeling the source domain. Ghahramani and Jordan (1995) addressed the use of the EM algorithm for training generative mixture model classifiers from labeled source data and unlabeled target data. However, this method aims to model all the samples in the source and target domains together, while for domain adaptation, it is assumed that the target data will only be drawn from the distribution $P_t(X)$. Tan et al. (2009) modified the relative contributions of source and target domains in EM. They increased the weight

on the target data at each iteration, while Dai et al. (2007b) specified the tradeoff between the source and target data terms by estimating KL divergence between the source and target distributions, with more weight on the target data as KL divergence increases. Saerens et al. (2002) fitted a generative model, where it was assumed that conditional distribution is the same between source and target domains but that the class proportions differ. EM was applied on the target domain only, where the source domain was used to estimate the initial model, and the M-step updated the model based on the class proportion counts in the target domain. Thus, this method has the ability to adapt the model from the source domain to the target domain. In (Pérez & Sánchez-Montañés 2007), EM is performed on the target data but with an additional term penalizing the distance between the new parameters and the source domain parameters. Ling et al. (2008b) applied the information bottleneck approach (Tishby et al. 2000) to a domain adaptation text classification problem. They categorized the unlabeled samples in order to maximize certain information theory objectives. In practice, this approach has a similar iterative implementation to EM, and results in a generative distribution over features for each category. Self-training methods have been applied to domain adaptation on Natural Language Processing (NLP) tasks including parsing (McClosky et al. 2006; Roark & Bacchiani 2003; Sagae 2010; Sagae & Tsujii 2007); part-of-speech tagging (Jiang & Zhai 2007a); conversation summarization (Sandu et al. 2010); entity recognition (Ciaramita & Chapelle 2010; Jiang & Zhai 2007a, 2007b); sentiment classification (Tan et al. 2008); spam detection (Jiang & Zhai 2007a); cross-language document classification (Rigutini et al. 2005; Shi et al. 2010); and speech act classification (Jeong et al. 2009).

Co-training (Blum & Mitchell 1998) is a semi-supervised learning method that has also been used for domain adaptation. The method is based on the idea of multi-view learning in which two different classifiers are trained based on different feature representations. Each classifier is used alternately to label new examples from the unlabeled pool. Nominated examples from each classifier are then used to train the opposite classifier on the next round. In Co-training, it does not explicitly assume

that $P_s(X) \neq P_t(X)$. For instance, Wan (2009) used it for cross-language sentiment classification; a machine translation system was used to derive versions of each document in both languages, representing two views. Wang (2009) used co-training to adapt parsers trained on newswire to other genres.

2.6.2.3 Feature Representation Methods

A Number of domain adaptation methods have changed the feature representation to better represent the shared characteristics of the two domains. These methods assume that certain features are domain-specific while others are domain-independent, or that mappings exist from the original feature space to a latent feature space that is shared between domains. The feature representation approaches can be categorized in two classes (Margolis 2011): (1) distribution similarity approaches aim explicitly to make the source and target domain sample distributions similar, either by penalizing or removing features whose statistics vary between domains (Arnold et al. 2007; Aue & Gamon 2005; Jiang & Zhai 2007b; Satpal & Sarawagi 2007) or by learning a feature space projection in which a distribution divergence statistic is minimized (Chen et al. 2009a; Pan et al. 2008; Pan et al. 2009); (2) Latent feature approaches aim to construct new features by analyzing large amounts of unlabeled source and target domain data (Blitzer et al. 2007b; Blitzer et al. 2009; Blitzer et al. 2006; Ciaramita & Chapelle 2010; Huang & Yates 2009, 2010a, 2010b; Pan et al. 2010).

Saptal and Sarawagi (2007) proposed a method to find features that minimize a distance between means of the two domains while simultaneously maximizing classification performance on the source-domain training data. They used conditional random fields, where the goal is to learn a weight vector w on features that are functions of both $\boldsymbol{x}$ and $\boldsymbol{y}$. The distance measure between domains is actually the sum of distances between sample means for each feature. However, since the features depend on y, which is unknown in the target domain, they take expected values using the $P\ (y|x, w)$. Since the feature mean estimation depends on w, learning follows an iterative procedure whereby the distances between features means are computed, the

weights w are updated; the weights are fixed, and the feature means are updated. Arnold et al. (2007) applied a similar idea for maximum entropy classifiers in which instead of penalizing features with large divergence, they scaled each feature in the source domain so that its expected value matched that in the target domain. Jiang and Zhai (2007b) used a regularized logistic regression classifier to allow the domain-independent features to be regularized less in training, compared to the domain-specific features. However, their method for finding the domain-independent features assumes that there are multiple source domains. Aue and Gamon (2005) and Margolis et al. (2010) applied domain adaptation approaches for sentiment classification and speech act classification respectively. They removed the source domain-specific features and then trained the model on reaming features; both found that the approach was successful at improving cross-domain performance in some cases, but that it degraded performance in others. Pan et al. (2009) and Chen et al. (2009) used Maximum Mean Discrepancy (MMD) (Borgwardt et al. 2006) , which is the distance between sample means in a Reproducing Kernel Hilbert Space, to measure the feature distribution difference. They aimed to explore a feature representation to minimize the feature distribution difference between source and target distributions. The methods proposed in (Chen et al. 2009; Pan et al. 2008; Pan et al. 2009) are based on MMD to maximize the similarity between source and target domains. These methods are similar to covariate shift approaches particularly the method proposed in (Huang et al. 2007) which is also based on MMD. These approaches aimed to tackle the domain adaptation problem by learning new feature representations, while the covariate shift method (Huang et al. 2007) is based on weighting training samples.

In domain adaptation scenarios, some features only appear in the source domain (source domain-specific) or target domain (target domain-specific) and some occur in both domains (domain-independent). If training takes place on original feature representation, which includes all features, using the source data, then the learning-based model cannot use the target domain-specific features. However, using unlabeled source and target data together we might be able to derive a new feature

representation that aggregates source domain-specific, target domain-specific and domain-independent features which behave similarly. Several methods have aimed to learn the set of feature weights which are used to linearly project the original feature space into a new feature space called latent feature space. Methods that derive such linear feature transformations include the Latent Semantic Analysis (LSA), Principle Component Analysis (PCA), Structural Correspondence Learning (SCL), and Canonical Correlation Analysis (CCA). Generally, these methods use observed feature co-occurrences in the unlabeled source and target samples to derive the new feature space (Margolis 2011).

LSA and PCA, which have been applied for unsupervised dimensionality reduction, use the Singular Value Decomposition (SVD) of the sample-feature matrix to compute a low-rank data representation. A variety of studies (Ando 2004; Huang & Yates 2009; Pan et al. 2010) have applied SVD to conduct experiments for domain adaptation in Natural Language Processing (NLP) problems. SCL was originally proposed by Ando and Zhang (2005) for semi-supervised learning. It applies Alternating Structural Minimization (ASM) to linearly project the original feature space. Blitzer et al. (2007a; 2007b; 2006) proposed the SCL algorithm to define a pivot feature on the target domain from both domains and then uses unlabeled instances from the target to create the classification model. A number of studies have applied SCL for domain adaptation in different machine learning applications: part-of-speech tagging (Blitzer, Dredze & Pereira 2007); sentiment classification tasks (Blitzer, McOnald & Pereira 2006); cross-language sentiment classification using machine translation (Prettenhofer & Stein 2010; Wei & Pal 2010); cross-language speech act classification (Margolis et al. 2010) conversation summarization (Sandu et al. 2010); and entity recognition (Ciaramita & Chapelle 2010). However, it is not effective for low-dimensional feature representations (Sandu et al. 2010) or for most kinds of cross-domain speech act classification (Margolis et al. 2010), where many important features were found to be mutually exclusive rather than correlated. Later, Tan and Cheng (2009) proposed a new version of SCL with some improvements

using feature and sample weighting. Recently Ji et al. (2011) proposed another version of SCL which trains separate prediction models on each domain. The experimental results demonstrate that it solves the problems of contradictory predictor features across domains. Blitzer et al. (2009) proposed a method to use CCA, which is an unsupervised dimensionality reduction method, for domain adaptation. It uses two views for each sample and forms a linear projection for each view into a new space, such that projections are maximally correlated.

2.6.2.4 CLUSTER-BASED LEARNING METHODS

The general assumption of these methods, which have been extensively applied in semi-supervised learning, is that two data points are likely to have the same label if there is a high density path between them (Gao et al. 2008). These methods aim to construct a graph in which the labeled and unlabeled samples are the nodes, with the edge weights among samples based on their similarity.

Xing et al. (2007) introduced a novel algorithm known as bridge refinement to modify the predicted labels of instances from a target domain. The authors used a mixture distribution of the training and test data as a bridge to transfer feature distribution from the source domain to the target domain. They trained a generic classifier such as SVM or Naive Bayes on the source data to predict the initial labels for the target instances. The initial labels were adjusted by applying a graph-based SSL algorithm similar to label propagation, whereby close neighbors are applied to change the labels of similar instances in target domain. This is done twice: first using the mixture of source and target domains and second using only the target data for refinement. Ling et al. (2008a) introduced a classification framework in which the objective function searched the consistency between in-domain supervision and the out-of-domain intrinsic structure. They aimed to find a cut of the graph that optimizes a function that is a combination of: (1) Spectral clustering cost function based on the data similarity matrix; (2) soft must-link constraints for the labeled source data; and (3) Spectral clustering cost function on the test data only. Gao et al. (2008) assumed

that there are multiple models trained on an out of-domain data set or on several such sets. They partitioned the decision space by clustering the target data and then integrating these models using weights based on how well each model corresponded with local partitions. The idea is that some models are better suited to certain regions, which can be detected by correspondence with local cluster boundaries in that region. Pan et al. (2010) proposed a feature clustering method based on the co-occurrence of features on target and source domains. They divided the features into two categories of domain-specific and domain-independent features and then used spectral clustering to create common feature clusters among domains. In their experiments on cross-domain sentiment classification, this method performs better than both SCL and LSA in many cases. Dai et al. (2007a) proposed a co-clustering based algorithm to propagate the label information across domains for document classification. Xue et al. (2008) proposed a cross-domain text classification algorithm known as TPLSA to integrate labeled and unlabeled data from different but related domains.

Transfer learning, particularly domain adaptation, which is a new machine learning and data mining framework, can be implemented in many novel applications, but most studies have been conducted in text classification and reinforcement learning and there is a lack of published novel applications of transfer learning in other areas (Yang 2009).

CHAPTER 3

ADAPTIVE INFERENCE-BASED FUZZY NEURAL NETWORK

3.1 INTRODUCTION

This chapter presents the development of a new prediction approach for bank failure prediction using FNN. The proposed approach contains three main phases:

(1) A pre-processing technique called Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al. 2002), aiming to deal with the imbalanced data-sets problem in failure prediction,

(2) A clustering technique and specifying the network structure and rule formulation algorithm to dynamically compute the input fuzzy clusters and fuzzy rules from numerical training data and

(3) An adaptive inference system with a parametric t-norm operator in the learning algorithm to reduce prediction error.

This prediction approach has improved failure prediction accuracy, which is an essential feature for construction of a FEWS, on a data-set from Federal Reserve Bank of Chicago. The results show that the proposed approach performs very competitively in comparison with three existing financial warning systems: GenSo-EWS (Tung et al. 2004); FCMAC-EWS (Ng et al. 2008); and MLP (Lin & Lee 1996), two popular fuzzy neural networks: ANFIS(Jang 1993); DENFIS (Kasabov & Qun 2002) and one rule learning algorithm: C4.5 (Batista et al. 2004). It also supplies a

valuable and comprehensive financial knowledge base. The novelty of the proposed approach not only organizes the appropriate phases together as a framework to establish a financial early warning system, but also presents a new NN structure with rule generation and learning algorithms to obtain better results. Moreover compared to related studies that have used FNN methods for failure prediction, but which have not handled imbalance data set problems, the proposed approach takes this problem and its effect into account and successfully solves it.

This paper is organized as follows: Section 3.2 introduces preliminaries including the class imbalanced data set problem and a related solution, proper measure to evaluate prediction accuracy, and a fuzzy clustering method. Section 3.3 outlines the proposed approach and presents the structure of the proposed FNN with its rule generation algorithm which is the second phase of the proposed approach. Section 3.4 presents an adaptive inference-based learning algorithm which is the third phase of the proposed approach. Section 3.5 explains the evaluation and analysis of the experimental results for the proposed approach. Finally, summary of this chapter are discussed in Section 3.6.

3.2 CONCEPTS AND DEFINITIONS

This section discusses the problem of imbalanced data sets, particularly in failure prediction, and outlines the pre-processing solution for the problem, which is generated from SMOTE. An appropriate technique to evaluate the accuracy is also presented and the Discrete Incremental Clustering (DIC) method for clustering financial data is addressed. These techniques will then be used in the following sections.

3.2.1 THE PROBLEM OF IMBALANCED DATA SET

Over recent years, the imbalanced data sets problem has demanded considerable attention in the field of classification and prediction (Chawla et al. 2004; YANG 2006). During learning from imbalanced data-sets, the classifier will obtain a high predictive accuracy for the majority class, but will predict poorly for the minority

class which is equally necessary in prediction (Weiss 2004). Likewise, the classifier may consider the minority class as noise, which is then ignored. Some research has shown that the imbalanced data-sets problem has a negative influence on the ability of most classification methods (Japkowicz & Stephen 2002; Orriols-Puig & Bernadó-Mansilla 2009).

3.2.2 The Synthetic Minority Oversampling Technique

There are a large number of techniques which have been proposed to deal with the imbalanced data-sets problem. These techniques can be categorized into three groups: (1) internal techniques that present a new, or modified, algorithm to cope with the imbalanced data-sets problem; (2) external techniques that use pre-processing steps to reduce the effect of this problem; and (3) solutions using both indicated directions simultaneously. The main advantage of external methods is that they are more independent and adaptable and can be used with any classifier. Some researches (Fernández et al. 2009a, 2009b; Fernández et al. 2008) show that over-sampling methods, particularly SMOTE (Chawla et al. 2002), significantly improve the prediction accuracy of Fuzzy Rule Based Classifier Systems. This technique enables the minority instances to be over-sampled by producing synthetic examples along the line segments joining any/all of the K minority examples nearest neighbors. To do this, synthetic examples are created by multiplying the distance between the sample under consideration and its nearest neighbor, by a random number between 0 and 1, and adding it to the considered sample. This approach effectively makes the decision region of minority class more general, and therefore the over-fitting problem is avoided and minority examples spread further into the majority samples.

3.2.3 Accuracy Evaluation

Prediction accuracy is evaluated based on a confusion matrix which addresses the number of correctly and incorrectly predicted samples for each class. The table TP

(TN) contains the number of instances that are predicted correctly positive (negative). FP (FN) is the number of examples that is predicted wrongly positive (negative), which actually belong to the negative (positive) class. The important point when dealing with the imbalanced data-sets problem is that, in comparison with the majority class, the minority class has very little impact on accuracy (3.1), which is used mostly in experimental measures.

$$Acc = \frac{TP+TN}{TP+FN+FP+TN}. \quad (3.1)$$

A more correct metric is presented instead of using accuracy:

$$GM = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{FP+TN}}, \quad (3.2)$$

where $\frac{TP}{TP+FN}$ is called sensitivity and $\frac{TN}{FP+TN}$ is called specificity, which measures the effectiveness of the prediction algorithm in any class. The proposed metric (3.2) is a geometric mean of sensitivity and specificity, because both are expected to be high simultaneously (Barandela et al. 2003; Kubat et al. 1998).

3.2.4 Discrete Incremental Clustering

FNN derives fuzzy rules from clusters. Performing a cluster analysis is important, particularly in the second type of FNNs described in Section 3.1, and is the first step towards modeling the problem. There is a novel self organizing clustering technique which outperforms other techniques; DIC method is a dynamic clustering technique avoiding drawbacks such as stability-plasticity and inflexibility found in other methods and computing trapezoidal-shaped fuzzy sets (Tung et al. 2004). These fuzzy sets are applied as $ITerm_{ij}, i \in \{1,2,\dots,n\}\ \&\ j \in \{1,\dots,k_i\}$ and $OTerm_{ij}, i \in \{1,2,\dots,m\}\ \&\ j \in \{1,\dots,k_i\}$ for defining input and output linguistic terms respectively in an FNN structure.

3.3 Fuzzy Neural Network Structure and Rule Generation Algorithm

In this section, first the proposed approach is outlined as shown in Figure 3.1. The proposed structure of FNN and its rule generation algorithm are described. The main idea of the rule generation algorithm is to first consider all possible rules in a problem domain and then different weights are assigned to these rules during the rule updating phase. According to weights, weak rules are pruned and only strong rules are used in inference and learning algorithms.

3.3.1 Fuzzy Neural Network Prediction Approach Outline

The proposed approach contains three main phases: Phase 1: Applying SMOTE to deal with the imbalanced data-sets problem in failure prediction. Phase 2: Using DIC method, proposing an FNN structure and developing a rule generation algorithm to dynamically compute the input fuzzy clusters and fuzzy rules from numerical training data. Phase 3: Developing an adaptive inference-based learning algorithm to reduce prediction error. This approach is shown in Figure 3.1.

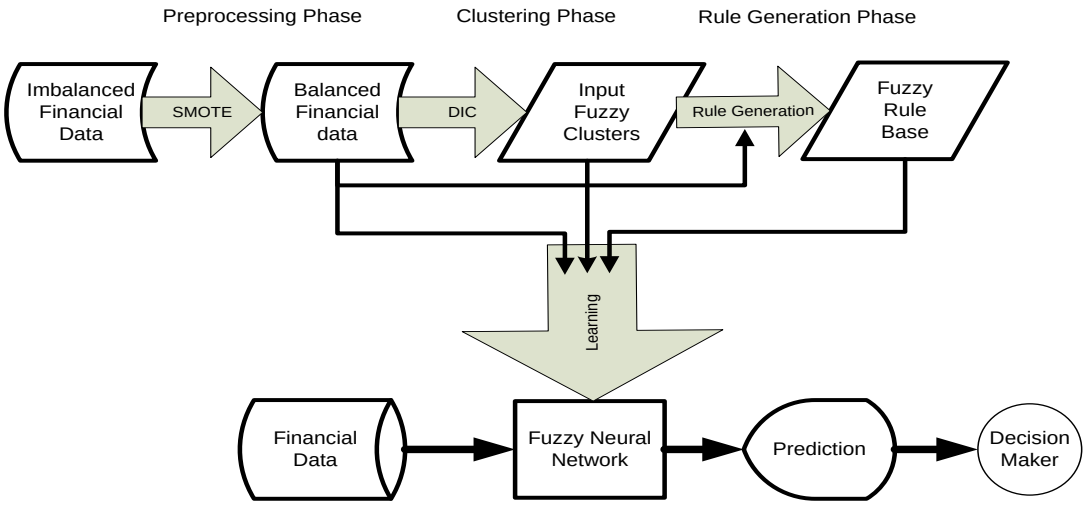


Figure 3.1: Outline of the Proposed Fuzzy Neural Network Prediction Approach

3.3.2 STRUCTURE OF FUZZY NEURAL NETWORK

Assume that $X = [x_1, x_2, \dots, x_n]$ and $Y = [y_1, y_2, \dots, y_m]$ represents the vector of inputs and outputs (calculated by network) respectively. In addition, assume that the vector $D = [d_1, d_2, \dots, d_m]$ represents the desired outputs (actual output) required during the learning phase. The proposed network has five layers of nodes: I) input layer consisting of input nodes $It_i, i \in \{1,2, \dots, n\}$, which have a single input x_i; II) input cluster layer including cluster nodes $ITerm_{ij}, i \in \{1,2, \dots, n\}$ & $j \in \{1, \dots, k_i\}$ which represent linguistic terms for each input. For instance, $ITerm_{23}$ represents the third linguistic term of input x_2; III) rule layer consisting of rule nodes $R_h, h \in \{1,2, \dots, l; l = \sum_{i=1}^{n} k_i \times \sum_{i=1}^{m} k_i\}$,which are representative of each rule and connects only one labels of each input (x_i ; $i \in \{1,2, \dots, n\}$) to one labels of each output ($y_i; i \in \{1,2, \dots, m\}$); IV) output cluster layer including cluster nodes $OTerm_{ij}, i \in \{1,2, \dots, m\}$ & $j \in \{1, \dots, k_i\}$, which represent linguistic terms for each output, for instance, $OTerm_{35}$ represents the fifth linguistic term of output y_3; V) output layer consisting of output nodes $Ot_i, i \in \{1,2, \dots, m\}$, which have single output y_i. Figure 3.2 shows the structure of the proposed FNN.

3.3.3 RULE GENERATION ALGORITHM

The rule generation algorithm is developed by formulating an algorithm which is based on the structure of the proposed FNN. In this algorithm, ISP_h is the set of terms for all input linguistic terms (layer II nodes) that contribute to the antecedent of rule node R_h and OSP_h refers to all output linguistic terms (layer IV nodes) that form the consequent of rule node R_h:

$ISP_h = \{ITerm_{1P_1}, ITerm_{2P_2}, \dots, ITerm_{nP_n}\}$ and

$OSP_h = \{OTerm_{1Q_1}, OTerm_{2Q_2}, \dots, OTerm_{mQ_m}\}$

if and only if

R_h : If x_1 is $ITerm_{1P_1}$ and x_2 is $ITerm_{2P_2}$ and ... and x_n is $ITerm_{nP_n}$

then y_1 is $OTerm_{1Q_1}$ and y_2 is $OTerm_{2Q_2}$ and ... and y_m is $OTerm_{mQ_m}$) .

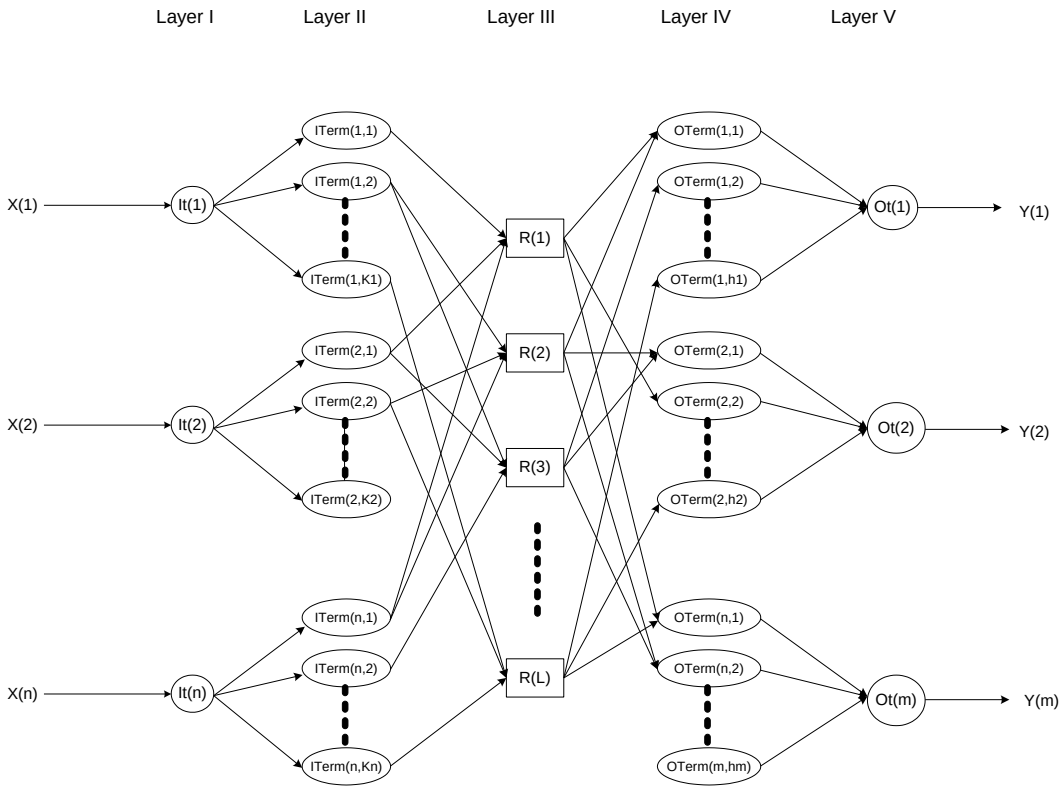


FIGURE 3.2: STRUCTURE OF PROPOSED FUZZY NEURAL NETWORK

<Rule Generation Algorithm>

The rule generation algorithm has three main steps as follows:

[Begin]

Step1: Firing input and desired vectors

The input vector $X(T) = [x_1(T), \dots, x_n(T)]$ and the desired output (actual output) vector $D(T) = [d_1(T), \dots, d_m(T)]$ are fed to the network through layer I and layer V respectively at T-th epoch. According to input and output linguistic terms (fuzzy sets) membership function, the membership value of inputs and outputs in each term is evaluated and the linguistic terms of each input and output with maximum membership value is selected:

$$MITerm(T) = \left\{ ITerm_{1b_1}, ITerm_{2b_2}, \dots, ITerm_{nb_n} | b_i = argmax\left\{ \mu_{ITerm_{ij}} \ (x_i(T)), j \in \{1, \dots, k_i\} \right\} \right\}, \quad (3.3)$$

$$MOTerm(T) = \left\{ OTerm_{1b_1}, OTerm_{2b_2}, \dots, OTerm_{mb_m} | b_i = argmax\left\{ \mu_{OTerm_{ij}} \ (d_i(T)), j \in \{1, \dots, k_i\} \right\} \right\}, \quad (3.4)$$

where k_i is the number of linguistic terms of i-th input or i-th output.

Step 2: Indicating nominated rule

In this step, the rule which has to be nominated for updating in each epoch is selected. The rule which satisfies (3.5) in epoch T-th is qualified for an update:

$$ISP_h = MITerm(T) \text{ and } OSP_h = MOTerm(T), \quad (3.5)$$

where h**,** m and n are the number of initial rules, outputs and inputs respectively, $h \in \{1,2, \dots, l\}$, $l = \sum_{i=1}^{n} k_i \times \sum_{i=1}^{m} k_i\}$.

Step 3: Rule updating

To update the rule weight in epoch T-th, (3.6) is applied:

$$W_h(T) = W_h(T-1) + \left(M_{ISP_h}(T) \times M_{OSP_h}(T) \right), W_h(0) = 0, h \in \{1,2, \dots, l\}, \quad (3.6)$$

where

$$M_{ISP_h}(T) = \mu_{ITerm_{1b_1}} \times \mu_{ITerm_{2b_2}} \times \dots \times \mu_{ITerm_{nb_n}}$$

and

$$M_{OSP_h}(T) = \mu_{OTerm_{1b_1}} \times \mu_{OTerm_{2b_2}} \times \dots \times \mu_{OTerm_{mb_m}}.$$

Some rules may be more important than others in the modeling of the problem domain. Also, there are insignificant rules which are created due to the existence of noisy training data. They may interfere with, or contribute errors to the network outputs. Therefore rule pruning is performed at the end of rule updating at each epoch to avoid these weak rules and also generating large number of rules. So the algorithm does not allow that the number of weak rules incrementally increases and instead it preserves the strong rules through the rule generation. Also it needs to mention that DIC algorithm includes a post processing step to integrate the obtained segments and reduce the number of segments and consequently number of rules significantly. To prune the weak rules a predefined threshold parameter $Thresh$ is considered and each

rule satisfying (3.7) will be pruned at the end of the updating step. By allocating a high value to the threshold, some significant rules may be eliminated and consequently, a portion of the significant input-output space of the problem will not be covered. If a low value is specified, a number of weak rules, which cause an increase in error and computational load, will remain.

$$l \times \left(\frac{W_h}{\sum_{k=1}^{l} w_k}\right) < Thresh \cdot \tag{3.7}$$

[END]

By using this rule generation algorithm, an initial consistent and compact fuzzy rule base is obtained. In order to gain more accurate prediction results, the rules are modified and updated. A learning algorithm will be presented to optimize the rule base in the next section.

3.4 Adaptive Inference-based Learning Algorithm

To enhance predictive accuracy, other adaptive neuro-fuzzy learning algorithms have modified parameters including input membership function (premise parameters), consequent parameters (S norm parameters) and rule weights during training, while the proposed algorithm modifies the t-norm parameter by applying a parametric t-norm (Dubois t-norm) in the inference system during learning. Those algorithms which need to change the parameters in the database during learning are practically expensive in respect of time and memory. To gain a more accurate prediction and cheaper performance, an adaptive inference-based learning algorithm is proposed to improve the prediction accuracy. It adjusts the parametric t-norm of the inference system and reduces the significance of rules causing errors and, therefore, augments accuracy. However, tuning other parameters in previous algorithms along with this parameter, which will form future work relating to this study, may even improve the performance significantly.

Assume that $X(T) = [x_1(T), \dots, x_n(T)]$ and $D(T) = [d_1(T), \dots, d_m(T)]$ are input and desired output vectors in T-th epoch of training respectively and there is a fuzzy rule base that includes l rules and each rule, R_i (i^{th} rule) has the form as follows:

R_i: If x_1 is $ITerm_{1j_1}$ ... and x_n is $ITerm_{nj_n}$ Then

$$y_1 \text{ is } OTerm_{1h_1} \dots \text{and } y_m \text{ is } OTerm_{mh_m} \cdot \tag{3.8}$$

<*Adaptive Inference-based Learning Algorithm*>

The proposed adaptive inference-based learning algorithm has seven steps as follows:

[Begin]

Step 1: Fuzzyfying

Input nodes act as singleton fuzzifier that fuzzyfy the crisp-valued inputs. Thus, the activation function of each node in layer I (Figure 3.2) is defined as:

$$Z_i^{(1)}(T) = f^{(1)}\big(x_i(T)\big) = \mu_{\tilde{x}_i}\big(\tilde{x}_i(T)\big) = f(x) = \begin{cases} 1, & \text{if} \quad \tilde{x}_i(T) = x_i(T) \\ 0, & \text{if} \quad Otherwise \end{cases}, \tag{3.9}$$

where $\tilde{x}_i(T)$ is the fuzzyfied equivalent of crisp input $x_i(T)$ and $Z_i^{(1)}(T)$ is the output of node $It_i, i \in \{1,2, \dots, n\}$ at T-th epoch.

Step 2: Antecedent matching

To perform antecedent matching of fuzzyfied inputs against linguistic terms, input cluster nodes compute the input membership value in each term. The activation function of each node in layer II (Figure 3.2) is defined as:

$$Z_{ij}^{(2)}(T) = f^{(2)}\left(Z_i^{(1)}(T)\right) = \mu_{ITerm_{ij}}\big(x_i(T)\big)$$

$$= \begin{cases} 0, & \text{if} \quad x_i \le l_{ij} \\ \frac{x_i - l_{ij}}{u_{ij} - l_{ij}}, & \text{if} \quad l_{ij} \le x_i \le u_{ij} \\ 1, & \text{if} \quad u_{ij} \le x_i \le v_{ij} \\ \frac{r_{ij} - x_i}{r_{ij} - u_{ij}}, & \text{if} \quad v_{ij} \le x_i \le r_{ij} \end{cases}, \tag{3.10}$$

where $Z_{ij}^{(2)}(T)$ is the output of node $ITerm_{ij}$ at T-th epoch and $i \in \{1,2, \dots, n\}$, $j \in \{1,2, \dots, k_i\}$ ·

Step 3: Rule fulfillment

To calculate the strength of activation of an antecedent for all rules with input vector $X(T)$, rule nodes compute the degree of rule fulfillment as their output. The higher the degree of fulfillment, the greater is the compatibility of the input to the antecedent of the rule. Hence, the activation function of each node, R_h ; $h \in \{1,2,\dots,l\}$, in layer III (Figure 3.2) is defined as:

$$Z_{R_h}^{(3)}(T) = f^{(3)}\left(Z_{(ij)_h}^{(2)}(T), i \in \{1,2,\dots,n\}\right) = t\left(Z_{(ij)_h}^{(2)}(T), i \in \{1,2,\dots,n\}\right), \quad (3.11)$$

where $Z_{(ij)_h}^{(2)}(T)$ is the output of the j-th term of the i-th input that is connected to the rule R_h and t is the conjunction operator. In order to adapt the inference system, this operator is considered as parameterized t-norm. Some researchers (Alcalá-Fdez et al. 2007; Marquez et al. 2007) have shown that a tuning parameter can significantly improve the accuracy of linguistic fuzzy systems. Furthermore, Dubois t-norm shown in (3.12) is used as a conjunction operator in this step because it is more efficiently computed, and because it provides better accuracy than other parametric t-norms (Alcalá-Fdez et al. 2007).

$$t_{Dubois}(x, y, \alpha) = \frac{x \times y}{\max(x, y, \alpha)}, \ \alpha \in [0,1]. \quad (3.12)$$

Dubois t-norm operates as the minimum and an algebraic product with $\alpha = 0.00$ and $\alpha = 1.00$ repectively. The activation function is represented as:

$$Z_{R_h}^{(3)}(T) = f^{(3)}\left(Z_{(ij)_h}^{(2)}(T), i \in \{1,2,\dots,n\}\right) = t_{Dubois}\left(Z_{(ij)_h}^{(2)}(T), i \in \{1,2,\dots,n\}, \alpha_h(T)\right)$$
$$= \left(\prod_{i=1}^{n} Z_{(ij)_h}^{(2)}(T)\right) / Max\left\{Z_{(ij)_h}^{(2)}(T), i \in \{1,2,\dots,n\}\right\}, \alpha_h(T)\Big), \quad (3.13)$$

where $Z_{R_h}^{(3)}(T)$ is the output of node R_h at T-th epoch.

In addition, $\alpha(0) = \left(\alpha_1(0), \alpha_2(0), \dots, \alpha_l(0)\right)$ is initialized as 1.00 at the beginning of the learning algorithm, and then, during the feedback learning phase, each α_h may be decreased separately.

Step 4: Consequent derivation

In this step, an aggregator operator is used to derive the consequent of fuzzy rules. The max operator which is considered as s-norm uses rule weights and rule outputs to compute the inferred output according to each rule in the rule base. Therefore, the activation function for each node in layer IV (Figure 3.2) becomes:

$$Z_{ij}^{(4)}(T) = f_{ij}^{(4)}\left(Z_{R_h}^{(3)}(T), h \in \{1, \dots, l\}, W_h\right) = Max\left(\prod_{h=1}^{l} W_h \times Z_{R_h}^{(3)}(T)\right), \quad (3.14)$$

where $Z_{R_h}^{(3)}(T)$ is the output of the h-th rule node in layer III that is connected to $OTerm_{ij}$ as its consequent.

Step 5: Defuzzifying

To defuzzify the derived fuzzy outputs, the weighted center of averaging technique (Lin & Lee 1996) is used. The activation function in layer V (Figure 3.2) is:

$$y_i(T) = Z_i^{(5)}(t) = f_i^{(5)}\left(z_{ij}^{(4)}(T), j = (1,2, \dots h_j)\right) = \frac{\sum_{j=1}^{h_j}(Z_{ij}^{(4)}(T) \times \tilde{m}_{ij})}{\sum_{j=1}^{h_j} Z_{ij}^{(4)}(T)}, \quad (3.15)$$

where $\tilde{m}_{ij} = \frac{u_{ij}+v_{ij}}{2}$ and $y_i(T)$ is the output of i-th node Ot_i in layer V.

Step 6: Error evaluation

During the feed forward phase, the input vector $X(T)$ is presented to the network and the output vector $Y(T)$ results in T-th epoch. At the beginning of the feed backward pass, $Y(T)$ is compared against the desired output vector $D(T)$ and the resulting error is used to modify the vector $\alpha(T)$ in T-th epoch. Regarding the framework of imbalanced data sets, the geometric mean presented in (3.2) should be used to calculate the error for all outputs. The following equation denotes the error in the T-th epoch.

$$Error(T) = \left(1 - G(T)\right) = 1 - \sqrt{\frac{TP(T)}{TP(T) \times FN(T)} \times \frac{TN(T)}{FP(T) \times TN(T)}}, \quad (3.16)$$

where $G(T)$ is the accuracy of model till T-th epoch of training.

Step 7: Modifying t-norm

All links in the proposed network do not have a particular weight that affects the error in the feed backward phase. The first issue to be taken into consideration in order to achieve better accuracy is the significance of antecedents, which make inaccuracy, should be reduced. Dubois t-norm performs like minimum t-norm if the (3.17) becomes satisfied and Dubois t-norm approaches to product t-norm, which reduces the significance of the antecedent causing inaccuracy, by increasing $\alpha_h(T)$. Hence, better prediction can be achieved by increasing $\alpha_h(T)$ of rule R_h which participates in producing $Error(T)$:

$$Z_{(ij)_h}^{(2)}(T) \geq \alpha_h(T). \tag{3.17}$$

To modify the parameter $\alpha_h(T)$ in each iteration (3.18) is used:

$$\alpha_h(T+1) = \alpha_h(T) - \sigma \times (Error(T) \times W_h), \tag{3.18}$$

where σ is a positive experimental consistent less than 1.00. In conclusion the t-norm parameters $\alpha_h(T)$ are tuned during learning the FNN at each epoch (T) and each rule (h) has own parameter α_h which need to be adjusted as a local inference mechanism.

[END]

As described in Sections 3 and 4 the proposed approach uses financial records to predict bank status as failure or survival. In fact, it pre-processes financial data to make balanced data-sets, then uses the DIC method and rule generation algorithm to make input fuzzy clusters, and formulates a fuzzy rule base, respectively. It then applies the proposed learning algorithm to adjust the parameters, and predict the bank situation as accurately as possible.

3.5 Empirical Results Analysis

This section outlines, and explains, the experimental results of the proposed approach, which contains rule generation and learning algorithms. The results are excellent when compared with the performance of the other three FEWS: GenSo-EWS(Tung et al. 2004), FCMAC-EWS (Ng et al. 2008) , and MLP (Lin & Lee 1996) , two popular

FNNs: ANFIS(Jang 1993) and DENFIS(Kasabov & Qun 2002) and one rule learning algorithm: C4.5 (Batista et al. 2004).

3.5.1 DATA SETS

The financial data set and financial variables are extracted from Call Report Data, which is downloaded from the website of Federal Reserve Bank of Chicago[5] and the status of each bank is identified according to the Federal Financial Institutions Examination Council[6]. Two data sets, which include the observation period of the survived banks of 21 years from Jun 1980 to Dec 2000 and based on the history of each bank in FFIEC, are considered. There are 561 failed banks and 3285 survived ones in the first data-set provided by this study, and there are 548 failed banks and 2555 survived ones in the second data set presented by (Ng et al. 2008; Tung et al. 2004) . Although, Tung, Quek and Cheng (Tung et al. 2004) used nine financial variables, according to their statistical significance and correlation, It was observed that the model with three variables (indicated by *) instead of nine covariates has less created rules, less computational load and more prediction accuracy. Each covariate is ranked based on the importance of a feature as a result of a future selection process and three features with the highest grade are selected (Ng et al. 2008). The definitions of all variables are described in Table 2.1. The proposed approach is run by nine inputs and three inputs separately and the results are then compared.

3.5.2 RESEARCH DESIGN AND PRE-PROCESSING

To evaluate the proposed approach three different levels of experiments with two categories of variables, including six scenarios, are performed. Table 3.1 demonstrates these experiments. To examine these scenarios, two data sets are applied as shown in Tables 3.2 and 3.3. It should be mentioned that the number of records for experiments with nine variables is the same as for the three covariates. As

[5] **http://www.chicagofed.org**
[6] **http://www.ffiec.gov/nicpubweb/nicweb/NicHome.aspx**

shown in Table 3.2, the failed banks in first data set include, on average only 13.5% of the whole data, while survived banks consist of 86.5%. According to Table 3.3, 17.66% and 82.34% of the second data set are failed and survived banks respectively. This situation not only addresses the imbalanced data sets problem in both data sets, but also this problem is more severe in the first data set.

TABLE 3.1: RESEARCH DESIGN

Level	Scenario	Training data usage	Number of variables
Level 1	1	last available financial records	9
	2	last available financial records	3
Level 2	3	financial records one year prior to the last one	9
	4	financial records one year prior to the last one	3
Level 3	5	financial records two year prior to the last one	9
	6	financial records two year prior to the last one	3

TABLE 3.2: NUMBER OF AVAILABLE RECORDS IN DATA SET 1 FOR EACH SCENARIO

	Number of variables	Total Number banks	Number of Survived banks	Number of Failed banks
Last available record	9	3846	3285(85.41%)	561(14.59%)
	3			
One year prior	9	3725	3209(86.15%)	516(13.85%)
	3			
Two years prior	9	3593	3725(88.17%)	425(11.83%)
	3			

TABLE 3.3: NUMBER OF AVAILABLE RECORDS IN DATA SET 2 FOR EACH SCENARIO

	Number of variables	Total Number banks	Number of Survived banks	Number of Failed banks
Last available record	9	3103	2555(82.34%)	548(17.66%)
	3			
One year prior	9	3046	2572(84.44%)	474(15.56%)
	3			
Two years prior	9	2943	2585(87.84%)	358(12.16%)
	3			

To evaluate the impact of this problem and reduce its influence, the results of data with two different pre-processing methods are compared with the results of imbalanced data. The first method down-sampled training sets by randomly pruning away redundant survived banks until the number of survived and failed banks is equal. The second method applies the SMOTE technique described in Section 3.2 to training data-sets. The number of failed banks increases to the number of survived ones to achieve a balanced data set, which improves the accuracy of prediction without losing important information.

In each level, the data set splits into two pools: (1) failed banks denoted with output 1; (2) survived banks denoted with output 0, and there are five cross-validation groups: CV1, CV2, CV3, CV4 and CV5 which include 20% and the remaining 80% of both pools randomly, to form the training set and testing set, respectively. The training set of the five groups are mutually exclusive. The proposed network is trained using training data-sets and evaluated by the testing data-sets of all five cross-validation groups.

3.5.3 Experiment using Different Scenarios

First nine and then three covariates of the training data-sets are used as inputs to feed into the network. Outputs of the network range from 0.00 to 1.00 and the classification threshold, varying from 0 to 1, is used to distinguish between failed and survived banks. As a result different values for GM used to investigate the performance of the network are achieved. As shown in Figures 3.3 to 3.6, GM gets maximum value at a definite threshold which is considered the optimum accuracy of the network in each cross-validation group of each data-set, which is balanced by two mentioned methods. The threshold of five cross-validation groups of two data-sets for two scenarios (scenarios 1 and 6) with SMOTE as preprocessing method is shown in these figures.

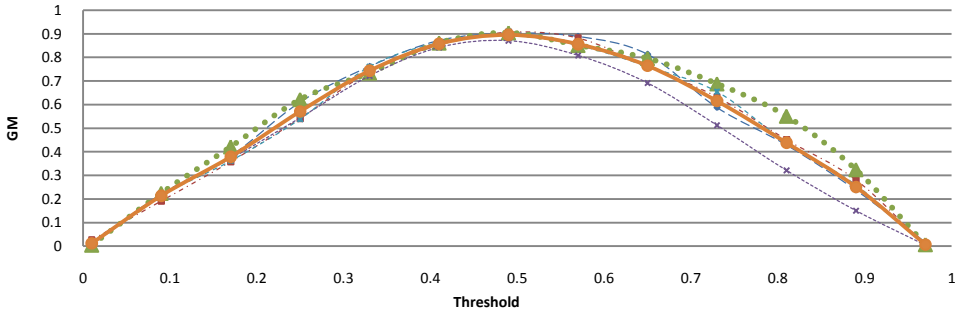


FIGURE 3.3: DATA SET 1, SMOTE, SCENARIO 1

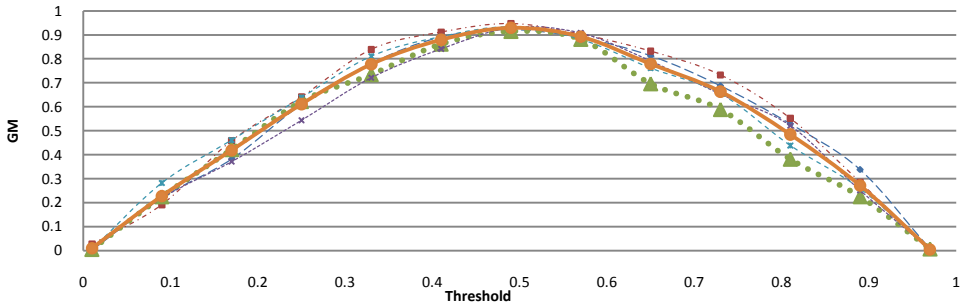


FIGURE 3.4: DATA SET 1, SMOTE, SCENARIO 6

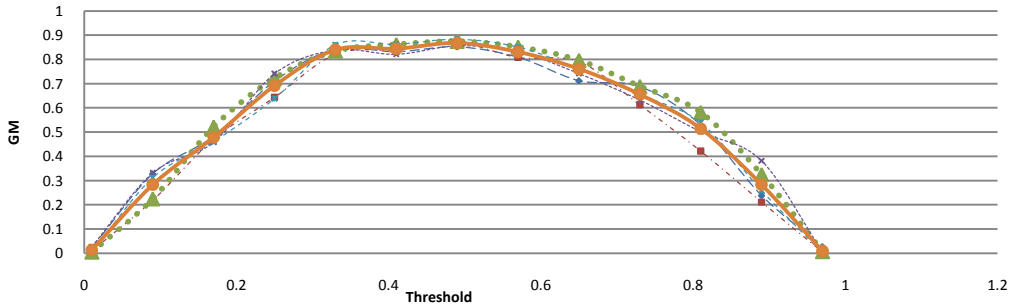


FIGURE 3.5: DATA SET 2, SMOTE, SCENARIO 1

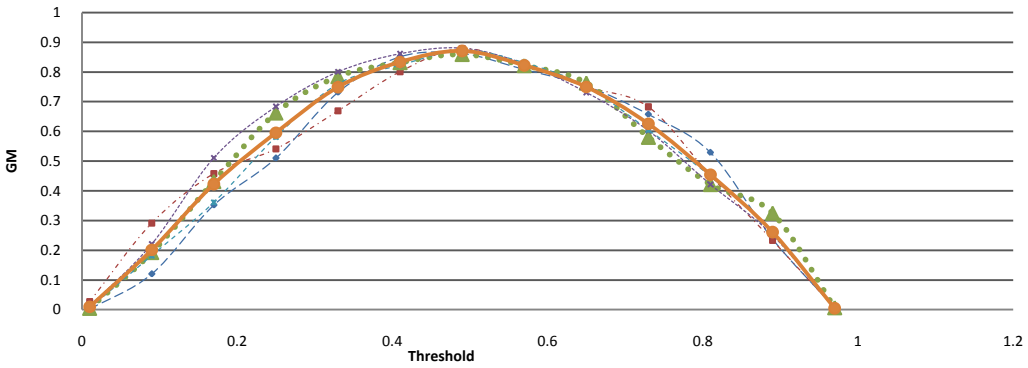


FIGURE 3.6: DATA SET 2, SMOTE, SCENARIO 6

The mean value and standard deviation for these experiments are summarized in Table 3.4. According to this table, there are a number of major issues which should be taken into account. The accuracy of the proposed approach is highest when SMOTE is applied as a pre-processing method. The accuracy, when Down-sampling is utilized, is located in the second position and the performance on imbalanced data is lowest in all scenarios when the same data sets are applied. For instance, in Scenario3-Data set 2, the accuracy is improved by 2% when SMOTE is applied. Table 3.5 shows the ranking of these categories, which is computed by the Friedman Aligned (Zar 1999) test. To simplify the forthcoming analyses only the results of cases in which SMOTE is applied will be considered in the comparisons.

Figure 3.7 overlay compares the performance of the proposed algorithm on all scenarios using average accuracy on both data sets in each scenario. As shown in Figure 3.7, the improvement of network performance from scenarios with nine variables to scenarios with three covariates in all experiments is considerable. For instance, the accuracy increases 3.48 % from scenario 1 to 2, 3.37 % from scenario 3 to 4 and 0.68 % from scenario 4 to 5.

TABLE 3.4: RESULT OF PROPOSED APPROACH FOR 6 SCENARIOS, 2 DATA SETS AND 2 PRE-PROCESSING METHODS

Scenario	Data-set	Pre-processing Method	CV1	CV2	CV3	CV4	CV5	Std deviation	Mean value
Scenario 1	Data set 1	null	88.97	92.49	90.39	92.37	91.1	1.46	91.06
		Down-sampling	91.88	95.15	90.71	91.96	94.45	1.88	92.83
		SMOTE	93.25	95.07	90.43	92.01	94.34	1.85	93.02
	Data set 2	null	93.21	90.47	91.09	89.15	92.33	1.58	91.25
		Down-sampling	93.63	92.27	91.34	90.51	91.67	1.16	91.88
		SMOTE	94.24	93.31	94.59	93.06	94.28	0.67	93.9
Scenario 2	Data set 1	null	96.56	96.84	95.2	94.28	92.97	1.61	95.17
		Down-sampling	95.34	94.44	96.01	98.39	97.5	1.60	96.34
		SMOTE	95.15	95.67	97.43	97.01	98.34	1.30	96.72
	Data set 2	null	96.61	93.67	99.13	95.28	94.13	2.20	95.76
		Down-sampling	99.28	95.55	93.12	95.22	96.43	2.24	95.92
		SMOTE	96.07	95.33	97.81	98.06	98.59	1.40	97.17
Scenario 3	Data set 1	null	88.14	86.07	86.49	85.15	89.25	1.65	87.02
		Down-sampling	86.38	88.28	87.43	90.03	86.53	1.50	87.73
		SMOTE	90.15	90.67	90.43	87.01	89.34	1.49	89.52
	Data set 2	null	86.92	87.12	88.67	87.34	86.36	0.85	87.28
		Down-sampling	88.36	86.95	87.08	88.25	86.59	0.81	87.45
		SMOTE	89.36	90.41	89.92	90.81	89.22	0.68	89.94
Scenario 4	Data set 1	null	91.73	90.64	89.02	90.25	91.79	1.14	90.68
		Down-sampling	90.11	91.38	91.67	92.9	92.01	1.02	91.61
		SMOTE	92.24	94.71	91.56	93.11	93.22	1.19	92.97
	Data set 2	null	90.91	91.82	91.26	89.46	90.41	0.90	90.77
		Down-sampling	90.9	91.66	88.49	92.43	91.61	1.51	91.02
		SMOTE	92.59	93.09	92.48	94.67	93.33	0.88	93.23
Scenario 5	Data set 1	null	82.21	83.62	85.76	84.39	83.5	1.30	83.89
		Down-sampling	84.25	85.55	87.05	84.19	86.13	1.23	85.43
		SMOTE	85.23	87.77	86.32	85.02	87.34	1.23	86.34
	Data set 2	null	84.16	83.33	84.93	83.56	85.09	0.79	84.21
		Down-sampling	85	84.82	84.04	83.92	85.61	0.70	84.68
		SMOTE	87.79	86.14	86.9	87.05	87.28	0.60	87.03
Scenario 6	Data set 1	null	84.21	83.44	86.38	86.05	85.63	1.26	85.14
		Down-sampling	85.3	87.11	87.13	85.24	87.66	1.13	86.49
		SMOTE	86.23	87.9	85.87	88.02	87.34	0.98	87.07
	Data set 2	null	85.27	84.12	85.94	87.1	85.66	1.08	85.62
		Down-sampling	84.36	85.6	86.48	86.93	87.06	1.12	86.09
		SMOTE	87.76	88.92	87.09	87.51	86.97	0.78	87.65

TABLE 3.5: FRIEDMAN RANKING TO EVALUATE THE PREPROCESSING METHODS

Algorithm	Ranking
SMOTE	6.50
Down-sampling	18.75
Null	30.25

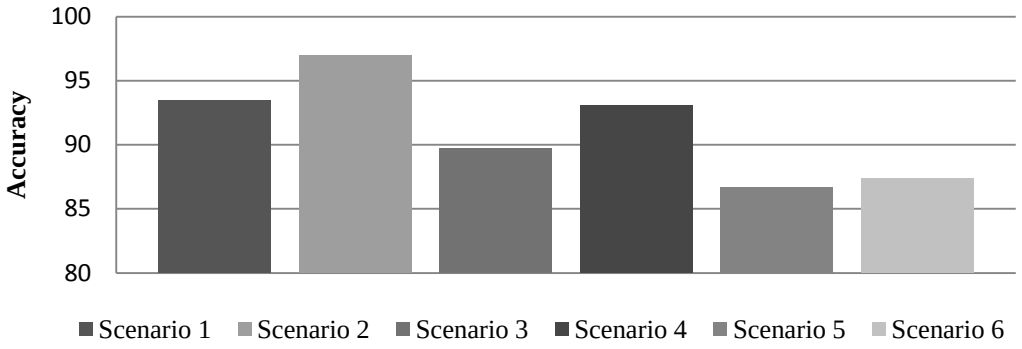


FIGURE 3.7: AVERAGE ACCURACY OF ALL SCENARIOS ON BOTH DATA SETS

To justify this significant difference the Holm test (Holm 1979), which is a non-parametric statistical tests for multiple comparison, has been applied to six cases shown in Table 3.6. The results are represented in Table 3.7. As can be seen, because the p-value is less than its corresponding α, the null hypothesis is rejected and a meaningful difference in accuracy is proven in both levels of significance $\alpha = 0.05$ and $\alpha = 0.1$. To perform the statistical tests in this study MultiTest software (Demsar 2006; Garcia et al. 2010; Garcia & Herrera 2008), which can be downloaded from web page[7], has been applied.

TABLE 3.6: THE SIX CASES WHICH ARE USED FOR COMPARISON

Data set	Three Variables	Nine Variables
Last available data set1	93.02	96.72
Last available data set2	93.9	97.17
one year prior data set1	89.52	92.97
one year prior data set2	89.94	92.23
two years prior data set1	86.34	87.07
two years prior data set2	87.03	87.65

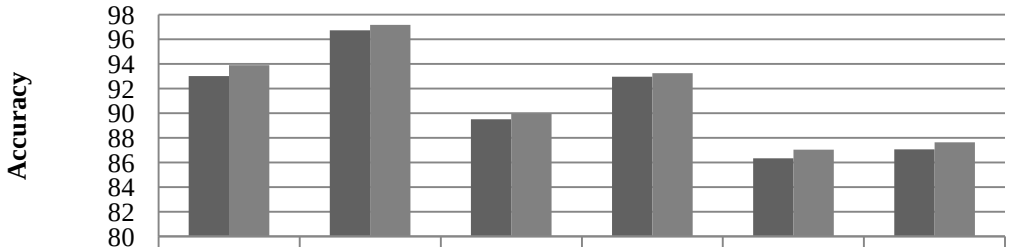
TABLE 3.7: HOLM'S TEST FOR COMPARISON OF 3-VARIABLE SCENARIOS WITH 9-VARIABLE SCENARIOS

Level of Significance	Hypothesis	$z = (R_0 - R_1)/SE$	p-value	α
$\alpha = 0.05$	3-variable scenarios vs. 9-variable scenarios	2.44949	0.014306	0.05
$\alpha = 0.1$	3-variable scenarios vs. 9-variable scenarios	2.44949	0.014306	0.1

[7] http://sci2s.ugr.es/sicidm

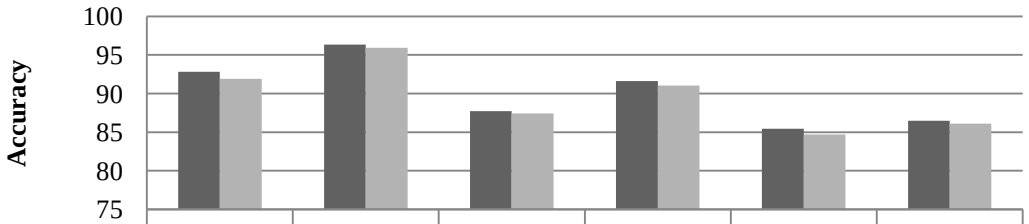
However, the proposed approach is more accurate with three variables, but this difference drops as it is transformed from classifiers to an early warning system as a predictor of bank status for the next one or two years. Likewise, scenarios using three variables consider fewer features and consequently, they generate less knowledge (fuzzy rules) which can be comprehensively used to describe the financial situation of banks and make timely decisions to prevent failure. Therefore, in spite of the fact that the proposed approach has better accuracy with three variables, the scenarios with nine covariates are recommended due to their better knowledge generation ability.

Moreover, according to Figure 3.8, the accuracy of prediction model on data set 1 is less than the accuracy on data set 2 when SMOTE is applied in all scenarios. One of the reasons for this difference is the imbalance problem. As mentioned in Section 5.2, data set 1 is more imbalanced than data set 2 and so the accuracy on data set 1 is less than on data set 2. As shown in Figure 3.9, the accuracy of the prediction model using data set 1 with down-sampling method is more than when using data set 2 with the same method. The reason is that the size of the training data set 1 is bigger than that of the training data-set 2 after down sampling.



	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Data Set 1	93.02	96.72	89.52	92.97	86.34	87.07
Data Set 2	93.90	97.17	89.94	93.23	87.03	87.65

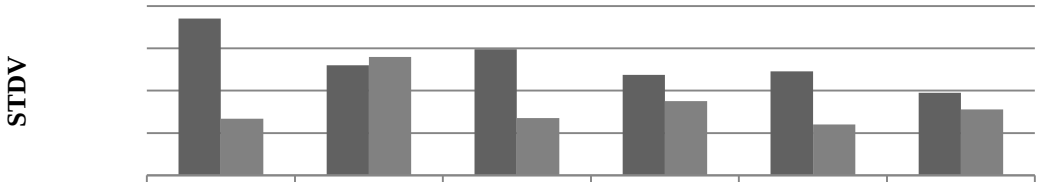
FIGURE 3.8: ACCURACY FOR TWO DATA SETS USING SMOTE IN ALL SCENARIOS



	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Dataset 1	92.83	96.34	87.73	91.61	85.43	86.49
Dataset 2	91.88	95.92	87.45	91.02	84.68	86.09

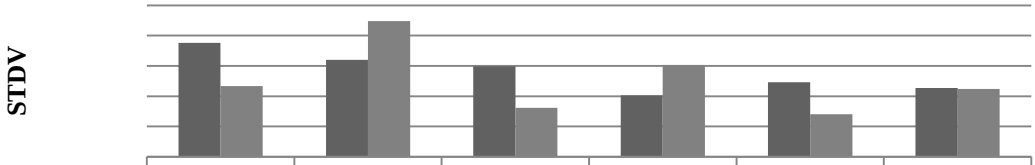
FIGURE 3.9: ACCURACY FOR TWO DATA SETS USING DOWN-SAMPLING IN ALL SCENARIOS

Likewise, Figures 3.8 and 3.9 demonstrate that there is a reduction in the accuracy of the proposed approach from level 1 (last available records) to level 3 (two years prior) in two data-sets using two pre-processing methods. While, Figures 3.10 and 3.11 show that the proposed approach surprisingly performs more consistently in prediction rather than classification. For instance, the standard deviation in scenarios using nine variables with SMOTE method on data-set 1 has a decline of 0.36 and 0.27 from level 1 to 2, and from level 2 to 3, respectively. The same can be said for scenarios with three variables: the standard deviation reduces from 1.30 to 1.19, and then to 0.97. It can be concluded that, although it is expected to have more uncertainties with the results of level 3 where the prediction is two years ahead from the available data, rather than level 1 where the network actually classifies banks into two groups using the last available records, the results represent less uncertainties and the approach has more consistent performance if it is applied as predictor.



	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Dataset 1	1.85	1.30	1.49	1.19	1.23	0.98
Dataset 2	0.67	1.40	0.68	0.88	0.60	0.78

FIGURE 3.10 : STANDARD DEVIATION FOR TWO DATA SETS USING SMOTE IN ALL SCENARIOS



	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Dataset 1	1.88	1.60	1.50	1.02	1.23	1.13
Dataset 2	1.16	2.24	0.81	1.51	0.70	1.12

FIGURE 3.11: STANDARD DEVIATION FOR TWO DATA SETS USING DOWN-SAMPLING IN ALL SCENARIOS

3.5.4 BENCHMARK AGAINST OTHER MODELS

The proposed model is compared with five prediction models including MLP neural networks GenSo-EWS, FCMAC-EWS, ANFIS, DENFIS. The neural networks are back-propagation trained MLP networks (Lin & Lee 1996). They have 9-10-1 and 3-5-1 layer structures that have been empirically determined to provide optimal results for scenarios with nine covariates and scenarios with three variables, respectively. GenSo-EWS and FCMAC-EWS, which are introduced in (Tung et al. 2004) and (Ng et al. 2008) respectively, use fuzzy neural network to predict bank failure. ANFIS (Jang 1993) and DENFIS (Kasabov & Qun 2002) are two of the most popular fuzzy neural networks that have represented good results in previous studies. The results of these models and the proposed approach are summarized in Table 3.8, which demonstrates that the proposed approach outperforms other models in all scenarios using two data-sets and two pre-processing methods, except the MLP neural network. Likewise, although by applying the imbalanced data set (indicated by null in preprocessing column) the proposed model is less accurate than when the pre-processed data is used, the performance of the proposed model is significantly better than that of all other five models. In the following sections these experiments are discussed in detail.

TABLE 3.8: ACCURACY OF FIVE PREDICTION MODELS AND PROPOSED APPROACH

Scenario	Data-set	Pre-processing Method	GenSo-EWS	FCMAC-EWS	DENFIS (offline)	ANFIS	Proposed approach	MLP
Scenario 1	Data-set 1	Null	87.33	88.06	88.49	89.32	91.06	89.81
		Down-sampling	89.77	90.22	91.04	91.43	92.83	94.42
		SMOTE	89.32	90.88	91.38	92.09	93.02	94.59
	Data-set 2	Null	87.93	88.21	88.75	89.55	91.25	90.12
		Down-sampling	89.05	90.08	90.78	91.07	91.88	93.25
		SMOTE	91.13	92.51	92.88	93.26	93.9	95.03
Scenario 2	Data-set 1	Null	88.45	90.12	91.59	92.71	95.17	93.64
		Down-sampling	91.12	95.38	95.72	96.08	96.34	97.75
		SMOTE	90.44	95.92	96.01	96.26	96.72	98.79
	Data-set 2	Null	89.15	91.84	93.72	94.39	95.76	94.19
		Down-sampling	89.82	94.81	95.49	95.52	95.92	96.14
		SMOTE	91.65	96.28	96.31	96.68	97.17	98.8
Scenario 3	Data-set 1	Null	84.37	85.01	85.5	86.26	87.02	86.18
		Down-sampling	85.92	87.11	87.08	87.27	87.73	92.46
		SMOTE	86.17	87.39	87.87	88.63	89.52	93.08
	Data-set 2	Null	84.71	85.36	86.14	86.51	87.28	86.95
		Down-sampling	85.13	86.69	86.65	86.81	87.45	91.52
		SMOTE	86.69	87.82	88.38	89.03	89.94	94.57
Scenario 4	Data-set 1	Null	82.41	87.41	88.39	90.13	90.68	89.86
		Down-sampling	83.26	89.63	90.27	90.66	91.61	93.25
		SMOTE	85.04	91.31	91.56	92.83	92.97	94.19
	Data-set 2	Null	82.76	88.24	89.05	90.3	90.77	90.26
		Down-sampling	83.47	89.45	89.58	90.51	91.02	92.64
		SMOTE	87.15	91.79	91.83	92.91	93.23	95.41
Scenario 5	Data-set 1	Null	77.08	79.81	80.53	82.06	83.89	82.27
		Down-sampling	79.26	82.61	83.11	84.36	85.43	86.2
		SMOTE	81.25	83.55	84.29	85.48	86.34	86.41
	Data-set 2	Null	77.92	81.43	82.37	82.46	84.21	82.57
		Down-sampling	80.02	82.29	82.71	83.56	84.68	85.03
		SMOTE	81.83	85.28	85.36	85.65	87.03	89.1
Scenario 6	Data-set 1	Null	76.11	79.92	81.42	83.16	85.14	83
		Down-sampling	78.42	83.54	84.08	85.81	86.49	88.3
		SMOTE	79.14	85.22	85.51	86.28	87.07	88.74
	Data-set 2	Null	77.16	81.01	82.41	83.86	85.62	83
		Down-sampling	78.27	83.69	83.73	85.31	86.09	88.06
		SMOTE	77.94	86.18	86.34	86.96	87.65	89.21

Furthermore the proposed approach using SMOTE is benchmarked against C4.5 as rule learning algorithm using SMOTE-ENN (Batista et al. 2004). This approach is well known to tackle the imbalanced classification problem. SMOTE-ENN is proposed by Batista et al. (2004) to apply Wilson's Edited Nearest Neighbor Rule (ENN) to the over sampled training sets as a data cleaning method. The algorithm was run using KEEL software (Alcal´a-Fdez et al. 2009) with the recommended parameter values given in this platform. Table 3.9 shows the results of this comparison using twelve case studies. It demonstrates that the proposed approach outperforms the C4.5 in all case studies. However this comparison is also examined statistically in the following section.

TABLE 3.9: ACCURACY OF C4.5 WITH SMOTE-ENN AND PROPOSED APPROACH WITH SMOTE

Scenario	Data set	C4.5 with SMOTE-ENN	Proposed Approach with SMOTE
Scenario 1	1	91.88	93.02
	2	93.45	93.9
Scenario 2	3	96.37	96.72
	4	96.52	97.17
Scenario 3	5	88.11	89.52
	6	88.63	89.94
Scenario 4	7	92.38	92.97
	8	92.71	93.23
Scenario 5	9	85.09	86.34
	10	85.66	87.03
Scenario 6	11	85.70	87.07
	12	86.55	87.65

3.5.4.1 BENCHMARK USING PRE-PROCESSED DATA

In this section the performance of the proposed approach is benchmarked against that of the models for the twelve case studies, including six scenarios per data set where SMOTE is applied. First, a post hoc statistical analysis is applied to show the difference in algorithm's performance. Table 3.10 shows the ranking, which is computed by a Quade test (Conover 1999). The best ranking is obtained by the MLP, then the proposed approach, and ANFIS.

TABLE 3.10: QUADE RANKING FOR ALL ALGORITHMS USING BALANCED DATA (SMOTE)

Algorithms	Ranking
MLP	0.99
Proposed approach	1.99
ANFIS	3.00
DENFIS	3.99
FCMAC-EWS	5.00
GenSo-EWS	6.00

To clarify the results, the Holm test is applied for comparison by considering a level of significance $\alpha = 0.05$. Table 3.11, which is associated with the Holm procedure, shows all computations. In this table the algorithms are ordered with respect to the z-value obtained. The normal distribution is applied to gain the corresponding p-value associated with each comparison. Then it is compared with the associated α-Holm in the same row of the table to show whether the corresponding hypothesis of equal mean accuracy is rejected in favor of the best ranking algorithm or not (marked as Not Rejected). The tests reject all hypotheses of equity of mean accuracy for all algorithms except the proposed approach, compared with the MLP. Accordingly, it can be concluded that there is not a significant difference between the accuracy of the proposed approach and MLP. However it is worth mentioning that the proposed approach has knowledge generation capability, which is very important for an applicable method in the finance industry, whereas MLP does not. The MLP functions as a black box and knowledge solicitation from its trained structure is nearly impossible and it fails to provide knowledge about the reasons behind the prediction.

TABLE 3.11: HOLM TEST FOR COMPARISON OF ALL ALGORITHMS USING BALANCED DATA (MLP IS THE CONTROL MODEL, $\boldsymbol{\alpha = 0.05}$)

Algorithms	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Hypothesis
Gen-So-EWS	6.546	5.889E-11	0.01	Rejected for MLP
FCMAC_EWS	5.237	1.630E-7	0.0125	Rejected for MLP
DENFIS	3.928	8.568E-5	0.017	Rejected for MLP
ANFIS	1.619	0.009	0.025	Rejected for MLP
Proposed Approach	1.309	0.190	0.05	Not Rejected

The performance of the proposed approach is considered as control method and compared with all other FNN models by applying the Holm test in a level of

significance $\alpha = 0.05$. As can be seen from the computations in Table 3.12, all hypotheses, except the one corresponding to ANFIS, are rejected in favour of the proposed approach. It is concluded that the proposed approach significantly outperforms GenSo, FCMAC and DENFIS. The proposed approach and ANFIS are compatible and the difference in accuracy is not significant when SMOTE is applied as a pre-processing method.

TABLE 3.12: HOLM TEST FOR COMPARISON OF ALL FNNS USING BALANCED DATA (THE PROPOSED APPROACH IS THE CONTROL MODEL, $\boldsymbol{\alpha = 0.05}$)

Algorithms	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Hypothesis
Gen-So-EWS	6.197	5.763E-10	0.0125	Rejected for the proposed approach
FCMAC_EWS	4.647	3.358E-6	0.017	Rejected for the proposed approach
DENFIS	3.098	0.002	0.025	Rejected for the proposed approach
ANFIS	1.549	0.121	0.05	Not Rejected

To investigate the real difference among methods for which Holm test Not Rejected the hypothesis, another non-parametric statistical test is carried out to benchmark their performances. Wilcoxon Signed Rank test has been applied in a level of significance $\alpha = 0.05$ and the results are represented in Table 3.13. Wilcoxon tests are performed using twelve case studies which used in previous tests. According to the results for both comparisons: proposed approach vs. ANFIS and MLP vs. proposed approach, the null hypotheses are rejected in favor of PA and MLP respectively with 95% of confidence. Since the accuracy of PA and MLP are higher than the benchmarked methods in all 12 cases, Wilcoxon test strongly rejects both hypotheses.

TABLE 3.13: RESULTS OF TWO PAIRED SAMPLE WILCOXON SIGNED RANK TEST FOR COMPARISON OF PROPOSED APPROACH WITH ANFIS AND MLP USING BALANCED DATA

Hypothesis	Signed-Rank Statistic	E($W+$) Signed-Rank Score	Var($W+$), Variance of Score	Signed-Rank Z-Score	One-Sided p-value	Two-Sided p-value
Proposed Approach vs. ANFIS & MLP	78	39	162	3.0594117	0 .0011	0 .0022

Another notable issue, which can be extracted from the results, is the improvement of the approach's performance regarding the time period of the prediction when it is

compared with other FNNs. The mean accuracy of each algorithm in each pair of scenarios, which correspond to the same time period prediction, is calculated when SMOTE is applied in both data sets 1 and 2. Figure 3.12 represents the difference in accuracy of the proposed approach (indicated PA) against other FNNs when the prediction time period increases. It is clear that the gap becomes wider when models are applied to predict bank failure one and two years ahead. According to the increasing trend shown in Figure 3.12, it can be concluded that the superiority of the proposed approach is more significant as the time period of prediction becomes longer.

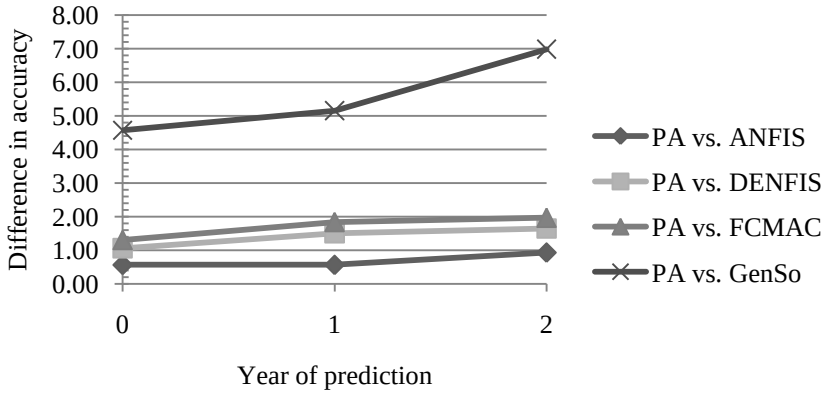


FIGURE 3.12: THE TREND OF DIFFERENCE IN THE PROPOSED APPROACH ACCURACY AGAINST OTHER FNNS ACCURACY

Finally the performance of the proposed approach using SMOTE is compared with C4.5 using SMOTE-ENN by Holm test. As it is showed in Table 3.14, the null hypothesis is rejected on these case studies. Accordingly the proposed approach produces significantly superior performance in a level of significance $\alpha = 0.05$.

TABLE 3.14: HOLM TEST FOR COMPARISON OF C4.5 USING SMOTE-ENN WITH PROPOSED APPROACH USING SMOTE

Level of Significance	Hypothesis	$z = (R_0 - R_1)/SE$	p-value	Conclusion
$\alpha = 0.05$	C4.5 with SMOTE-ENN vs. Proposed Approach with SMOTE	3.4641	5.32E-4	Rejected for the proposed approach

3.5.4.2 BENCHMARK USING IMBALANCED DATA

This section examine the performance of the proposed approach and compares it to the other five models when training data is imbalanced, as appearing in Tables 3.3 and 3.2. The benchmarks are performed by using twelve case studies which include six scenarios per data set. First, the Quade test is applied to gain the ranking of algorithms according to their accuracy. Table 3.15 demonstrates the ranking and shows that the proposed approach obtains the best ranking, with ANFIS and DENFIS achieving second and third positions respectively. Based on the ranking it can be concluded that the proposed approach outperforms all five models in dealing with imbalanced data.

TABLE 3.15: QUADE RANKING FOR ALL ALGORITHMS USING IMBALANCED DATA

Algorithms	Ranking
Proposed approach	1.00
ANFIS	2.35
MLP	2.64
DENFIS	4.00
FCMAC-EWS	5.00
GenSo-EWS	6.00

The Holm procedure is applied to evaluate the algorithms performance in more detail. The proposed approach is considered as a control model and compared with the other five models. All computations for the Holm test in the level of significance $\alpha = 0.05$ and $\alpha = 0.1$ are represented in Table 3.16. The tests reject all hypotheses of equality of the mean accuracy for all algorithms compared with the proposed approach in the level of significance $\alpha = 0.1$. The same situation applies for the level of significance $\alpha = 0.05$ except ANFIS. Based on the results, an interesting conclusion can be extracted. Although the accuracy decreases when the imbalanced data is used for training, the proposed approach outperforms other FNNs and even MLP. One of the reasons for this improvement is the application of parametric t-norms in inference and adjusting them during the learning. Likewise, employing an error function based on GM in the learning algorithm is another reason for its superiority. This reason is analyzed in the Section 3.5.4.3.

TABLE 3.16: HOLM TEST FOR COMPARISON OF ALL ALGORITHMS USING IMBALANCED DATA (THE PROPOSED APPROACH IS THE CONTROL MODEL)

Algorithms	$z = (R_0 - R_i)/SE$	p-value	Holm $\alpha = 0.05$	Hypothesis	Holm $\alpha = 0.10$	Hypothesis
Gen-So-EWS	6.546	5.889E-11	0.01	Rejected	0.02	Rejected
FCMAC_EWS	5.237	1.630E-7	0.0125	Rejected	0.025	Rejected
DENFIS	3.928	8.568E-5	0.0167	Rejected	0.033	Rejected
ANFIS	1.964	0.049	0.025	Not Rejected	0.05	Rejected
MLP	1.964	0.049	0.05	Rejected	0.1	Rejected

3.5.4.3 GM_ERROR FUNCTION ANALYSIS

To justify the contribution of the GM_Error in improving the overall performance, theoretically and experimentally the GM_Error function is compared with the RMSE error function, which is one of the most popular error functions in learning algorithms. Two issues need to be mentioned prior to the theoretical justifications:

(1) The proposed algorithm task is a prediction and has nominal integer outputs that indicate the class which each pair of input data belongs to;

(2) The more the learning error function is sensitive to false negative errors, the more the learning tuning is efficient because the false negative (FN) error has remarkable importance in overall system performance under the imbalance problem circumstances.

To simplify, a two-class prediction (classification) problem is analyzed and the approach can be extended for n class problems ($n > 2$). The RMSE in each epoch can be rewritten according to the first issue:

$$RMSE(t) = \sqrt{\frac{\sum_{i=1}^{n_t}(D_i - Y_i)^2}{n_t}}, \tag{3.19}$$

where D_i is desired (actual) input and

$$D_i = \begin{cases} 0, & \text{if } X_i \text{ belongs to Negative class} \\ 1, & \text{if } X_i \text{ belongs to Positive class} \end{cases}$$

and Y_i is the calculated output and

$$Y_i = \begin{cases} 0, & \text{if } X_i \text{ belongs to Negative class} \\ 1, & \text{if } X_i \text{ belongs to Positive class} \end{cases}$$

and n_t is the number of data pairs in t-th epoch. Hence

$$(D_i - Y_i)^2 = \begin{cases} 0, & \text{if } D_i = Y_i \\ 1, & \text{if } D_i \neq Y_i \end{cases} \text{ and } \sum_{i=1}^{n_t}(D_i - Y_i)^2 = FP + FN.$$

So we have:

$$RMSE(t) = \sqrt{\frac{\sum_{i=1}^{n_t}(D_i - Y_i)^2}{n_t}} = \sqrt{\frac{FP+FN}{n_t}} \text{ and } GM\_erorr(t) = 1 - \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}}.$$

To compare the sensitivity of error functions to the FN error, the first and second order of differential of the error functions are calculated and compared when the FN error increases. The amount of data in positive class ($n_P = TP + FN$) and Negative class ($n_N = TN + FP$) is constant and $n_t = n_p + n_N$ and $n_p \ll n_N$:

$$\frac{\partial RMSE}{\partial FN} = \frac{1}{2\sqrt{n_t(FP+FN)}}. \tag{3.20}$$

$$\frac{\partial GM\_error}{\partial FN} = \frac{\sqrt{TN}}{2\sqrt{(n_N)(n_P)(n_P - FN)}}. \tag{3.21}$$

$$\frac{\partial^2 RMSE}{\partial^2 FN} = \frac{-1}{4\sqrt{n_t(FP+FN)^3}}. \tag{3.22}$$

$$\frac{\partial^2 GM\_error}{\partial^2 FN} = \frac{\sqrt{TN}}{4\sqrt{(n_N)(n_P)(n_P - FN)^3}}. \tag{3.23}$$

It can be seen that $\frac{\partial RMSE}{\partial FN} < \frac{\partial GM_{error}}{\partial FN}$ which represents the GM_error is more sensitive to FN. Also, it is clear that $\frac{\partial^2 RMSE}{\partial^2 FN} < 0$ and $\frac{\partial^2 GM_{erorr}}{\partial^2 FN} > 0$ that addresses that by increasing FN the GM_error becomes more sensitive while RMSE becomes less. To illustrate, Figure 3.13 shows the error values and Figure 3.14 depicts the first order differential of one example in which $n_N = 1000, TN = 980, FP = 20, n_P = 100$ *and* FN is changing from 5 to 50.

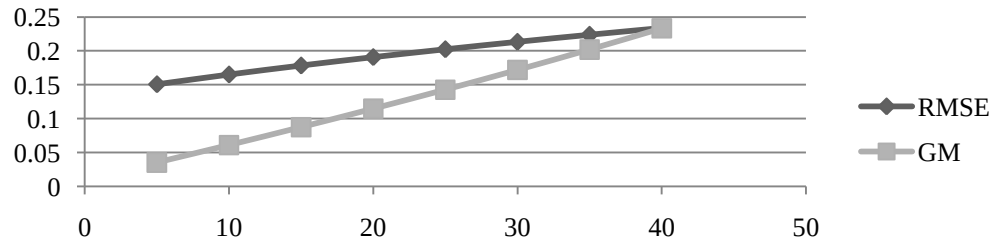


FIGURE 3.13: THE TREND OF RMSE AND GM_ERROR BY INCREASING THE FN

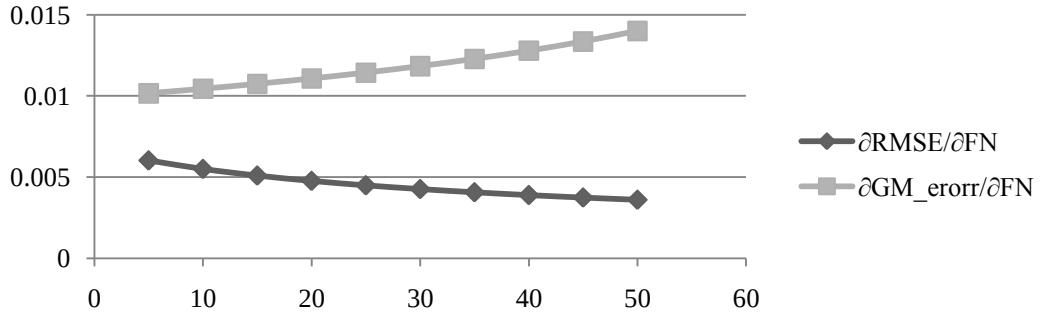


FIGURE 3.14: THE FIRST DIFFERENTIAL OF GM_ERROR AND RMSE BY INCREASING THE FN

Furthermore it is necessary to investigate the comparison of GM_error with F-Measure which is one of the popular evaluation measurements for learning algorithms. Obviously F-Measure is much more sensitive than G-Mean to the False Negative particularly when the data set is highly imbalanced. Accordingly F-Measure may seem to be more appropriate than G-Mean for the proposed FNN. However this high sensitivity to the False Positive error is not proper for training of the proposed FNN. As it is described in the Section 3.4, every single fuzzy rule has one specific t-norm parameter (α) which is adjusted during the training. The error computed by the error measurement (G-Mean) is used to adjust the parameters of rules including Positive and Negative rules which were fired and participate in causing the error in the underlying epoch. Since the number of instances in the training data set belong to the Negative class is much higher than that belong to Positive class, the majority of rules fired in each epoch are Negative rules. Accordingly the Negative rules, which produce FP error, inherently have much more effect than Positive rules on producing error. Then the computed error adjusts the parameters of the Positive and Negative rules similarly while they have not same contribution in producing error. To balance the influence of Positive and Negative rules and achieve more fair error in training of proposed FNN, G-Mean is applied, which is less sensitive to FP error and treat both classes equally, rather than F-Measure.

Similarly, some experiments are performed using imbalanced data sets and balanced data sets using SMOTE when GM_error, RMSE and F-Measure are applied in the algorithm. To demonstrate the impact of the GM_error and its contribution to the improved performance, the results are outlined in Table 3.17. As can be seen, the performance of the proposed approach is superior when using GM_error as the error function in learning algorithm particularly when it is applied on imbalanced data. To demonstrate this difference, Holm test is performed on the level of significance $\alpha = 0.05$ and the computations are represented in Table 3.18. All hypotheses are rejected in favor of GM_error and it can be concluded that the proposed algorithm performs better while GM_error is used in the learning algorithm.

TABLE 3.17: THE ACCURACY OF PROPOSED APPROACH WHEN GM_ERROR, RMSE AND F-MEASURE ARE APPLIED

Scenario	Data-set	Pre-processing Method	Error Functions		
			Gm_error	RMSE	F-Measure
Scenario 1	Data-set 1	null	91.06	89.23	90.33
		SMOTE	93.02	92.72	92.92
	Data-set 2	null	91.25	88.81	89.56
		SMOTE	93.9	92.91	92.63
Scenario 2	Data-set 1	null	95.17	93.38	93.80
		SMOTE	96.72	96.41	96.35
	Data-set 2	null	95.76	94.11	95.09
		SMOTE	97.17	96.55	96.41
Scenario 3	Data-set 1	null	87.02	85.06	86.18
		SMOTE	89.52	88.53	88.44
	Data-set 2	null	87.28	86.17	86.64
		SMOTE	89.94	89.49	89.73
Scenario 4	Data-set 1	null	90.68	88.59	89.91
		SMOTE	92.97	92.43	92.39
	Data-set 2	null	90.77	89.63	90.35
		SMOTE	93.23	92.77	92.68
Scenario 5	Data-set 1	null	83.89	82.91	83.49
		SMOTE	86.34	85.65	85.75
	Data-set 2	null	84.21	82.41	83.87
		SMOTE	87.03	86.69	86.49
Scenario 6	Data-set 1	null	85.14	83.28	83.93
		SMOTE	87.07	86.92	86.70
	Data-set 2	null	85.62	84.18	85.11
		SMOTE	87.65	87.09	87.36

TABLE 3.18: HOLM TEST TO COMPARE THE ACCURACY USING GM_ERROR AND RMSE

Preprocessing	Hypothesis	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Conclusion
Null	GM_error vs. RMSE	4.899	9.634 E-7	0.0167	Rejected in favor of GM_error
	GM_error vs. F-Measure	2.449	0.0143	0.025	Rejected in favor of GM_error
SMOTE	GM_error vs. RMSE	3.266	0.001	0.025	Rejected in favor of GM_error
	GM_error vs. F-Measure	4.082	4.458E-5	0.167	Rejected in favor of GM_error

3.5.5 FUZZY RULE ANALYSIS

The proposed approach formulates an intuitive Fuzzy Rule Base from numeric training data. This Fuzzy Rule Base can describe the inherent relationships between nominated financial covariates and their influence on the financial situation of observed banks. In comparison with the MLP neural network, which functions as a 'black box' from which it is nearly impossible to extract knowledge from its trained structure, the proposed approach provides valuable interpretable knowledge to support managers, regulators and banking analysts to make important decisions about the future of an institution. Because the decision making process aims to avoid failure and is important part of a useful and practical FEWS (which also needs a comprehensive knowledge in failure problem), the knowledge generation ability of the proposed approach is a remarkable advantage. This is one of the main reasons to apply a fuzzy inference system to the neural network structure.

As explained in Section 3.3, the fuzzy sets, or labels of nine or three selected financial covariates as input and created by DIC method and two fuzzy outputs called survived bank and failed bank will be used to describe the derived rules. Fuzzy sets and their linguistic terms depicted in Figure 3.15 for data-set 1, the cross-validation group CV1 and scenario1when using SMOTE illustrate this step. The fuzzy rules and their weights are extracted from the trained proposed network simply by tracing the

connections between nodes. All rules can be classified into two groups: positive rules identifying failed banks, and negative rules denoting survived banks. Table 3.19 lists five positive rules with the highest weight as well as five strong negative rules that compose approximately forty percent of the total rule firing frequency. As can be seen from this Table, the fuzzy rule base derived straight from the numerical financial training data is intuitive, is comprehensible and interpretable, and can be easily understood and employed by human users. For instance, this Table reveals that the majority of failed banks are not able to generate profits from investments (Low Return on equity), have to set aside a high proportion of capital for unrecoverable loans (Provision rate), and have poor capability in absorbing losses (Low Capital ratio). Financial analysts and regulators can apply synthetic data to simulate different scenarios to survey the occurrence of several situations and analyze the sensibility of the model, to define appropriate input values which lead a financial corporation to become stable, ongoing healthy organization.

3.6 SUMMARY

Although many statistical and soft computing models and methods have been applied to FEWS, they can not explicitly explain the implicit and intrinsic relationship between financial covariates and failure phenomena. They also suffer from different deficiencies which make them inapplicable. This study proposes a prediction approach to classify and predict bank failure more accurately, as well as to generate useful knowledge describing the influence of selected financial variables of failure. Through thirty six conducted experiments with two data sets and three preprocessing methods comparing with five very popular prediction models, the results have demonstrated that the proposed approach remarkably improves prediction accuracy and outperforms almost five other prediction models particularly on imbalanced data. Since knowledge is the fundamental base of a decision making framework, the rule formulation and knowledge creation ability of the proposed approach makes it a practical and useful component of a FEWS in the finance industry.

TABLE 3.19: FUZZY RULES DERIVED FROM THE FUZZY NEURAL NETWORK IN THE PROPOSED APPROACH

Type and number of rule	Description of rule	Weight %
Positive: Rule 1	If **Return on equity** is *low* Then the bank is failed bank.	11.36
Positive: Rule 2	If **Provision rate** is *high* Then the bank is failed bank.	10.12
Positive: Rule 3	If **Capital ratio** is *low* Then the bank is failed bank.	8.47
Positive: Rule 4	If **Provision rate** is *medium* AND **Capital ratio** is *slightly low* AND **Return on equity** is *medium* Then the bank is failed bank.	7.22
Positive: Rule 5	If **Provision rate** is *medium* AND **Capital ratio** is *medium* AND **Return on equity** is *medium* AND **Loan loss allowance** is *low* AND **Past due loan measure** is *medium* AND **Liquidity measure** is *high* Then the bank is failed bank.	6.50
Total		43.67
Negative: Rule 1	If **Provision rate** is *low* AND **Capital ratio** is *medium* AND **Return on equity** is *medium* AND **Loan loss** is *slightly low* AND **Past due loan measure** is *low* AND **Liquidity measure** is *low* AND **Non-interest profit** is *slightly low* AND **Loan growth** is *medium* AND **Net-interest margin** is *medium* Then the bank is survived bank.	12.53
Negative: Rule 2	If **Provision rate** is *low* AND **Capital ratio** is *medium* AND **Return on equity** is *medium* AND **Loan loss allowance** is *medium* AND **Past due loan measure** is *low* AND **Liquidity measure** is *low* AND **Non-interest profit** is *slightly low* AND **Loan growth** is *medium* AND **Net-interest margin** is *medium* Then the bank is survived bank	10.09
Negative: Rule 3	If **Provision rate** is *low* AND **Capital ratio** is *medium* AND **Return on equity** is *medium* AND **Loan loss allowance** is slightly *low* AND **Past due loan measure** is *low* AND **Liquidity measure** is *low* AND **Non-interest profit** is *low* AND **Loan growth** is *medium* AND **Net-interest margin** is *medium* Then the bank is survived bank	9.28
Negative: Rule 4	If **Provision rate** is *low* AND **Capital ratio** is *medium* AND **Return on equity** is *medium* AND **Loan loss allowance** is *medium* AND **Past due loan measure** is *low* AND **Liquidity measure** is *low* AND **Non-interest profit** is *low* AND **Loan growth** is *medium* AND **Net-interest margin** is *medium* Then the bank is survived bank	6.15
Negative: Rule 5	If **Provision rate** is *low* AND **Capital ratio** is *high* AND **Return on equity** is medium Then the bank is survived bank	5.19
Total		43.24

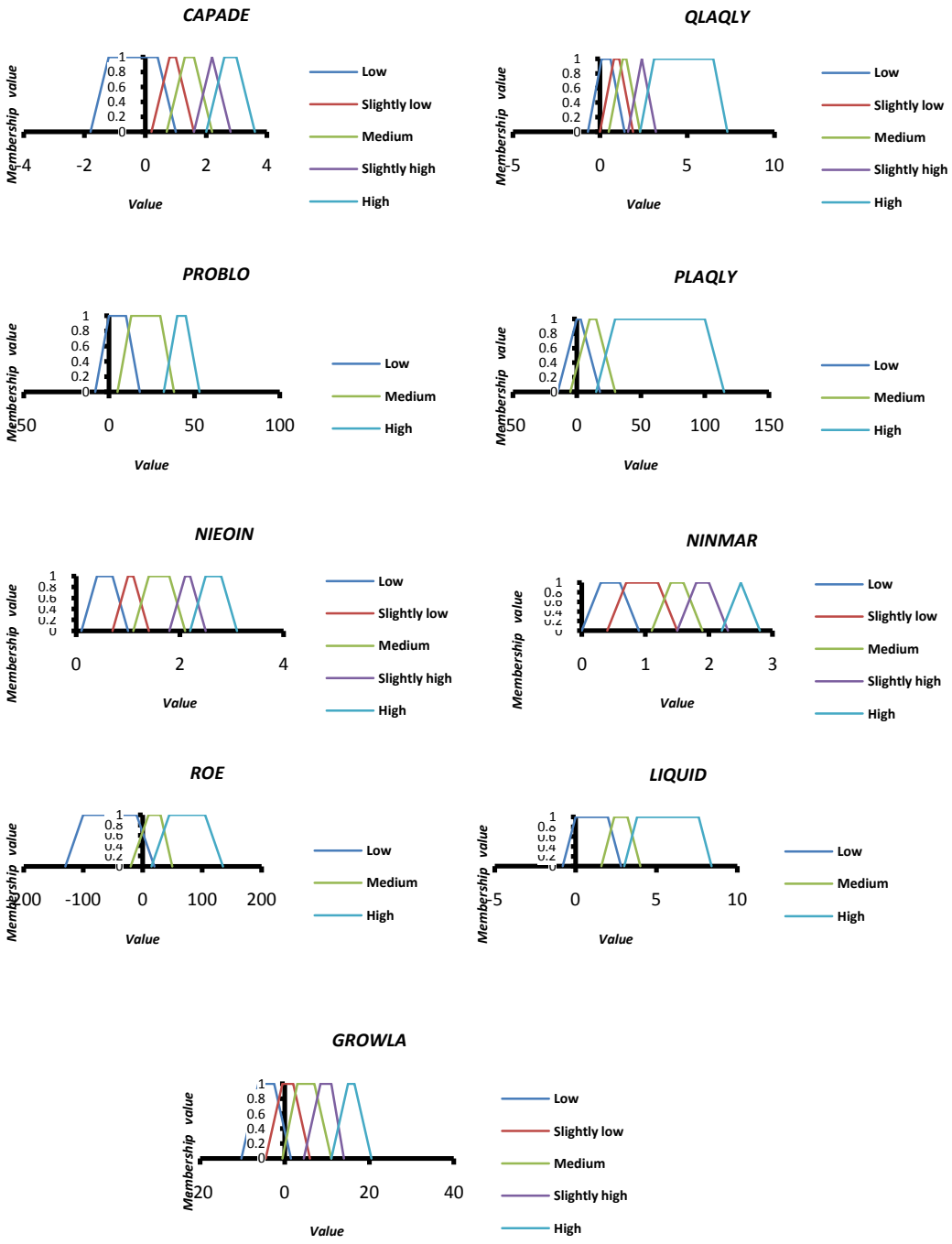


FIGURE 3.15: FUZZY SETS CALCULATED BY DIC ON CV1 GROUP IN SCENARIO 1

CHAPTER 4

MULTI STEP FUZZY BRIDGE REFINEMENT DOMAIN ADAPTATION

4.1 INTRODUCTION

Traditional machine learning models such as FNN have gained remarkable attention among researchers in a variety of computational fields, including that of prediction. However, most only work well under a common assumption that the training and the test data sets have the same feature space and the same distribution. As a result, when the distribution or feature space of the test data changes, the prediction models need to be rebuilt and retrained from scratch using newly collected training data. In real world applications, the feature space of the test data often changes. For example, more and more labeled financial data become out-of-date over time and new financial data may not follow the same distribution. Hence, past labeled data cannot be used to reliably predict the current financial situation of an organization. Additionally, collecting new training data and retraining a particular model is very expensive, and often practically impossible. Therefore, it would be very practical and profitable if the data collected from different time periods, or domains, could be utilized to assist current learning tasks, such as prediction.

Many domain adaptation methods have emerged in research with the aim of handling this issue. These methods assume that the feature spaces of both domains are similar but the marginal probability distribution of the data is different. However, most existing research in domain adaptation use probabilistic models which work well under statistical assumptions, but may be violated in real world applications. Moreover, they are not able to tackle uncertain values of real world problems and consequently decline in performance. Fuzzy sets and rough sets are more flexible toward these assumptions and are capable of handling the uncertainty, but there is no study that uses these soft computing techniques for transfer learning. Likewise, most of the existing transfer learning methods aim to refine the decision boundary and models, which makes these models highly complex computationally and dependent on the prediction model. Focusing on the given test data and refining the obtained labels would be an acceptable approach that is less computationally complex and more independent of the given prediction model (Xing et al. 2007).

A novel fuzzy domain adaptation method called Multi-Step Fuzzy Bridged Refinement (MSFBR) is proposed in this chapter. The MSFBR algorithm applies fuzzy sets, to handle uncertainty and enhance predictive accuracy of fuzzy neural network as an example of a shift- unaware prediction model. It refines the predicted labels and focuses on currently given test data instead of modifying the decision boundary, which makes the algorithm less computationally complex and more independent of the prediction model. In particular, the algorithm applies multi-step label refinements in mixture domains towards target distribution to decline the influence of the shift-unaware prediction model on performance. Bridged refinement (Xing et al. 2007), which is the closest study to this research and used for benchmarking in the experiments, only considers the similarity of instances with crisp values through two steps. The proposed label refinement is performed simultaneously, based on the similarity and dissimilarity of mixture domains instances. These capabilities enable the proposed MSFBR algorithm to be more accurate and able to be

practically implemented in real world applications, particularly in financial businesses with huge databases.

To validate and evaluate the MSFBR algorithm, a challenging real world application, long-term bank failure prediction, is employed. The growing development of machine learning has led researchers to employ new methods for bank failure prediction. However, machine learning models assumes that the test data and training data have the same distribution which, consequently, results in low predictive accuracy when it is used for long-term prediction since data distribution changes over a longer period. The proposed MSFBR algorithm effectively handles this problem and has produced better performances than fuzzy neural network. It is the first scientific attempt to utilize the transfer learning algorithm for long-term financial prediction.

This chapter makes the following contributions: (1) The proposed MSFBR algorithm is the first fuzzy domain adaptation algorithm to tackle features with vague (fuzzy) values and can therefore handle uncertainty and achieve better performance; (2) The MSFBR algorithm gains more independence from the prediction model by focusing on given test data and modifying the labels of target instances according to the distribution of the mixture domains in a multi-step method; (3) The MSFBR algorithm introduces a fuzzy similarity/dissimilarity-based learning method as local learning to refine the predicted labels for domain adaptation. It shows that under some conditions the local learning can be an appropriate solution for domain adaptation and can therefore improve accuracy. (4) The MSFBR algorithm provides a solution for the long-term prediction problem, particularly in bank failure prediction, in which the feature distribution changes over time.

This chapter is organized as follows. In Section 4.2, preliminary concepts are provided. Section 4.3 presents the MSFBR algorithm. The bank failure experimental illustration and results analysis are described in Section 4.4. Section 4.5 concludes the paper.

4.2 CONCEPTS AND DEFINITIONS

In this section the definition of domain, task, transfer learning and domain adaptation are introduced. Also some notations and definitions that will be used throughout the chapter are introduced.

Definition 4.1 (Domain) A domain, which is denoted by $D = \{F, P(X)\}$, consists of two components:

(1) Feature space F; and

(2) Marginal probability distribution $P(X)$, where $X = \{x_1, \dots, x_n\} \in F$.

For example if the learning task is the bank failure prediction**,** χ is the financial ratios that are applied for prediction, X is the set of all instances (banks) and $P(X)$ is the marginal distribution of these instances. In general, if two domains are different, then they may have different feature spaces or different marginal probability distributions.

Definition 4.2 (Task) A task, which is denoted by $T = \{Y, f(\cdot)\}$, consists of two components:

(1) A label space $Y = \{y_1, \dots, y_m\}$; and

(2) An objective predictive function$f(\cdot)$ which is not observed and to be learned by pairs $\{x_i, y_i\}$.

The function $f(\cdot)$ can be used to predict the corresponding label, $f(x_i)$, of a new instance x_i. From a probabilistic viewpoint, $f(x_i)$ can be written as $P(y_i|x_i)$. In bank failure prediction example, which is a binary prediction task, y_i can be the label of failed or survived. More specifically, the source domain can be donated as $D_s = \{(x_{s_1}, y_{s_1}), \dots, (x_{s_n}, y_{s_n})\}$ where $x_{s_i} \in F_s$ is the source instance or bank in bank failure prediction example and $y_{s_i} \in Y_s$ is the corresponding class label which can be failed or survived for bank failure prediction. Similarly, the target domain can be denoted as $D_t = \{(x_{t_1}, y_{t_1}), \dots, (x_{t_n}, y_{t_n})\}$ where $x_t \in F_t$ is the target instance and $y_{t_i} \in Y_t$ is the corresponding class label and in most scenarios $t_n \ll s_n$.

Definition 4.3 (Transfer learning) Given a source domain D_s and learning task T_s, a target domain D_t and learning task T_t, transfer learning aims to improve the learning of the target predictive function $f_t(\cdot)$ in D_t using the knowledge in D_s and T_s where $D_s \neq D_t$ or $T_s \neq T_t$.

In the above definition, the condition $D_s \neq D_t$ implies that either $F_s \neq F_t$ or $P_s(X) \neq P_t(X)$. For example, in bank failure prediction example, this means that between a source banking system and a target banking system, either the financial features are different between the two domains, or the marginal distributions of banks are different. Similarly, the condition $T_s \neq T_t$ implies that either $Y_s \neq Y_t$ or $f_s(\cdot) \neq f_t(\cdot)$. For instance, it corresponds to situation that source banking system has binary class labels of failed and survived, whereas the target banking system has more than two class labels, or source prediction model and target prediction model classify the identical bank in different class labels. In addition, there are some explicit or implicit relationships among feature spaces off two domains which imply that the source domain and target domain are related. It needs to be mentioned that when target and source domain are the same ($D_s = D_t$) and their learning tasks are also the same ($T_s = T_t$), the learning problem becomes a traditional machine learning problem.

Definition 4.4 (Domain Adaptation) A category of transfer learning in which $T_s = T_t$ and $D_s \neq D_t$ which implies that either $F_t \neq F_s$ or $P_t(X) \neq P_s(X)$.
A distinction exists between supervised domain adaptation, which assumes some labeled data in the target domain, vs. unsupervised domain adaptation, which assumes only labeled data from the source domain and unlabelled data from the target domain. In this situation, no labeled data in the target domain are available while a lot of labeled data in the source domain are available. In addition, according to above definition, domain adaptation also can be divided into two cases: (1) The feature spaces between domains are the same ($F_t = F_s$), but the marginal probability distributions of the input data are different ($P_t(x) \neq P_s(x)$). (2) The feature spaces between the source and target domains are different ($F_t \neq F_s$). The first category is

called as domain adaptation and the second category is called as cross-domain adaptation.

4.3 FUZZY BRIDGED REFINEMENT DOMAIN ADAPTATION

This section is composed of Section 4.3.1 which describes and proves the related theory of the proposed MSFBR algorithm and, Section 4.3.2 which presents the MSFBR algorithm based on the explained theory and its implementation. The implementation details and settings to find the best performance of MSFBR are also explained in Section 4.3.2.

4.3.1 BRIDGED REFINEMENT-BASED THEORY

Bridged Refinement theory assumes that the conditional probability of a specified label C, given an instance d, does not vary between different distributions: $P_{D_s}(C|d) = P_{D_{s\cup t}}(C|d) = P_{D_t}(C|d)$ although the marginal probability of instance d ($P(d)$) varies. This is based on the fact that, if an identical instance appears in the target and the source domain, the predicted label should be the same. The more similar instances that are in the target domain, the more the probability is that they have the same label. This situation forms a mutual reinforcement relationship between instances in a target domain and source domain and can be used to correct the predicted labels. Not only is this assumption considered in this research, but also a complementary idea is applied. It is assumed that the more different the instances are in the target domain, the less is the probability is that they have the same label. For instance, in a two class problem, significantly dissimilar instances are located in the opposite classes while the significantly similar instances are located in the same class. In the other words, the similarity and dissimilarity between instances simultaneously indicates their class labels. However, the similarity and dissimilarity functions play an important role and need to be defined well enough for mapping the instances and then discriminating the instances accurately. Recently (Balcan et al. 2008; Wang et al.

2007) developed theories for good similarity and dissimilarity functions and gave sufficient conditions for the functions to allow one learn well. Hence, the definitions and conditions can be used to define similarity and dissimilarity functions such that there is a high probability that similar instances will have same labels and dissimilar instances will have different labels. We used the main proposed theory to construct our functions and the most similar and dissimilar instances to a target instance are then applied to modify the class label.

Labeled instances are presented by $z_i = (x_i, y_i)$, where $x_i \in X$ and $y_i \in \{-1, 1\}$. Similarity and dissimilarity are nonnegative functions as follows:

1) Similarity function: $S(x_i, x_j)$ where $x_i, x_j \in X$ and $S(x_i, x_j) \in [0, 1]$.
2) Dissimilarity function: $D(x_i, x_j)$ where $x_i, x_j \in X$ and $D(x_i, x_j) \in [0, 1]$.

Definition 4.5 (Wang et al. 2007) Let $z_a, z_b, z_c \in X \times \{-1, 1\}$, similarity S and dissimilarity D functions are strongly (ε, γ)- good for learning problem if at least $1 - \varepsilon$ mass probability of instances z satisfy:

$p(S(x_a, x_b) > S(x_a, x_c) \,|y_a = y_b, y_a = -y_c) \geq 0.5 + \gamma/2,$

$p(D(x_a, x_b) < D(x_a, x_c) \,|y_a = y_b, y_a = -y_c) \geq 0.5 + \gamma/2,$

where the probability is over random instances z_b, z_c.

This definition says that $S(D)$ is a good similarity (dissimilarity) function for a learning problem if most instances (at least $1 - \varepsilon$ mass probability) are on average at least $\boldsymbol{\gamma}$ more similar (dissimilar) to random instances $x_b(x_c)$ of the same (opposite) label than they are to random instances x_c (x_b) of the opposite (same) label.

Theorem 4.1 (Wang et al. 2007) If S and D are strongly (ε, γ)- good, then $\left(\frac{4}{\gamma^2}\right) \ln (\frac{2}{\delta})$ positive examples and $\left(\frac{4}{\gamma^2}\right) \ln (\frac{2}{\delta})$ are sufficient so that with the probability $\geq 1 - \delta$, the above algorithm produces a classifier $f(x)$ with error at most $\varepsilon + \delta$.

The theory represents that using the sufficiently large set of positive and negative instances and similarity or dissimilarity functions, the constructed classifier specify the label of given instances accurately (error $\leq \varepsilon + \delta$). According to the Definition 4.5, Theorem 4.1 and using the fuzzy concepts, most similar and dissimilar instances to a given instance $x_a \in D_t$ are defined as follows.

Definition 4.6 Let $x_a \in D_t = (X_t, \mu_t(x))$ and, S and D are strongly (ε, γ)- good similar and dissimilar functions. $\beta, \partial > 1/2$, the sets of most similar $KS\_\beta(x_a)$ and most dissimilar $KD\_\partial(x_a)$ instances to x_a be defined as follows:

$KS\_\beta(x_a) = \{x_i \in D_t \cup D_s | \, S(x_a, x_i) \geq \beta, S(x_a, x_{i+1}) \geq S(x_a, x_i), |KS_\beta| = K_\beta\}$,

$KD\_\partial(x_a) = \{x_i \in D_t \cup D_s | \, D(x_a, x_i) \geq \partial, D(x_a, x_{i+1}) \geq D(x_a, x_i), |KD_\partial| = K_\partial\}$,

where $\forall x_i \in KS\_\beta, \mu_{l=y_a}(f(x_i)) > \emptyset \geq 1 - (\varepsilon + \delta)$

and

$\forall x_i \in KD\_\partial, \mu_{l=-y_a}(f(x_i)) > \emptyset \geq 1 - (\varepsilon + \delta)$

and

$\forall x_i, \mu_{l=-y_a}(f(x_i)) + \mu_{l=y_a}((x_i)) = 1$.

It suggests that the instances with the high value of similarity and dissimilarity to the underlying instance have high membership value $(1 - (\varepsilon + \delta))$ in the same and opposite label respectively using the constructed classifier $f(x)$.

Example 4.1 This very simple example aims to demonstrate KS_β and KD_∂**.** Let $x_a \in D_t$, $y_a = 1$, S and D are strongly $(\boldsymbol{\varepsilon = 0.01, \gamma = 0.9})$-good similar and dissimilar functions, $\delta = 0.01$ and x_{b_i} *and* x_{c_i} are the instance examples in $D_s \cup D_t$ which are ranked based on their similarity and dissimilarity to x_a.

TABLE 4.1: SIMILARITY AND MEMBERSHIP VALUE OF A SET OF EXAMPLES

	x_{b_1}	x_{b_2}	x_{b_3}	x_{b_4}	x_{b_5}	x_{b_6}	x_{b_7}	x_{b_8}	x_{b_9}	$x_{b_{10}}$
$S(x_a, x_{b_i})$	0.63	0.68	0.73	0.82	0.84	0.9	0.92	0.92	0.93	0.95
$\mu_{y_{b_i}=1}(f(x_{b_i}))$	0.95	0.95	0.95	0.95	0.96	0.96	0.96	0.96	0.98	0.99

TABLE 4.2: DISSIMILARITY AND MEMBERSHIP VALUE OF A SET OF EXAMPLES

	x_{c_1}	x_{c_2}	x_{c_3}	x_{c_4}	x_{c_5}	x_{c_6}	x_{c_7}	x_{c_8}	x_{c_9}	$x_{c_{10}}$
$D(x_a, x_{c_i})$	0.68	0.72	0.76	0.83	0.86	0.91	0.93	0.94	0.96	0.98
$\mu_{y_{c_i}=-1}(f(x_{c_i}))$	0.95	0.95	0.96	0.97	0.97	0.98	0.98	0.98	0.99	0.99

If $\beta = 0.93$ and $\partial = 0.90$ then $KS\_0.93 = \{x_{b_9}, x_{b_{10}} | \mu_{l=y_a}(f(x_{b_i})) \geq 0.98\}$ and $K_\beta = 2$. $KD\_0.90 = \{x_{c_6}, x_{c_7}, x_{c_8}, x_{c_9}, x_{c_{10}} | \mu_{l=-y_a}(f(x_{c_i})) \geq 0.98\}$ and $K_\partial = 5$. Definition 4.7 defines the refined label, which is computed by applying the most similar and dissimilar instances to the target instance. This measure is used in the Step 4-3 of the proposed MSFBR algorithm to refine the labels. To simplify the equations, we use $\mu_{l=y}(x_{b_i})$ and $\mu_{l=y}(x_{c_i})$ instead of $\mu_{l=y}(f(x_{b_i}))$ and $\mu_{l=y}(f(x_{c_i}))$ in the rest of the paper.

Definition 4.7 Let $z_a = (x_a, y_a) \in D_t$, $z_{b_i} = (x_{b_i}, y_{b_i}) \in KS\_\beta, z_{c_i} = (x_{c_i}, y_{c_i}) \in KD\_\partial$, $\mu_{y=l}(x)$ is the membership value of instance x in class l computed by a prediction model and S and D are strongly $(\boldsymbol{\varepsilon}, \boldsymbol{\gamma})$ -good similar and dissimilar functions, we call:

$$\mu'_{l=y}(x_a) = \alpha\left(\frac{\sum_{i=1}^{K_\beta} S(x_a, x_{b_i})\left(\mu_{l=y}(x_{b_i}) - \mu_{l=y}(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} D(x_a, x_{c_i})\left(\mu_{l=y}(x_{c_i}) - \mu_{l=y}(x_a)\right)}{K_\partial} + \mu_{l=y}(x_a)\right) + (1-\alpha)\mu_{l=y}(x_a)$$

the refined membership value of instance x_a in class label y (RMV_y) where $0 < \alpha < 1$ is the tradeoff coefficient.

Example 4.2 This example aims to demonstrate Definition 4.7 using the samples of Example 4.1. If $\beta = 0.93$, $\partial = 0.90$ and $\alpha = 0.3$ then $x_{b_i} \in \{x_{b_9}, x_{b_{10}}\}$ and $x_{c_i} \in \{x_{c_6}, x_{c_7}, x_{c_8}, x_{c_9}, x_{c_{10}}\}$. It is assumed that the membership value of instance $z_a = (x_a, y_a = +1)$ in class +1 computed by a given prediction model is 0.50: $\mu_{l=+1}(x_a) = 0.50$. Then we have:

$$\mu'_{l=+1}(x_a) = \alpha\left(\frac{\sum_{i=1}^{2} S(x_a, x_{b_i})\left(\mu_{l=+1}(x_{b_i}) - \mu_{l=+1}(x_a)\right)}{2} - \frac{\sum_{i=1}^{5} D(x_a, x_{c_i})\left(\mu_{l=+1}(x_{c_i}) - \mu_{l=+1}(x_a)\right)}{5} + \mu_{l=+1}(x_a)\right)$$
$$+(1-\alpha)\mu_{l=+1}(x_a) = 0.75.$$

As it can be seen the refined membership value of instance x_a to class $+1$ increases from 0.5 to 0.75 for the given example. Since $y_a = +1$, the expected membership value of instance x_a in the positive class (EMV_+1) $\mu^E_{y_a=+1}(x_a) = 1$. Consequently

the difference between RMV_+1and EMV_+1 is less than that between unrefined membership value (UMV_+1) and EMV_+1:

$|\mu^E_{y_a=+1}(x_a) - \mu'_{l=+1}(x_a)| = 0.25 \leq |\mu^E_{y_a=+1}(x_a) - \mu_{l=+1}(x_a)|$=0.50.

Similarly, it is assumed that the membership value of instance $z_a = (x_a, y_a = +1)$ in class -1 computed by a given prediction model is 0.50: $\mu_{l=-1}(x_a) = 0.50$ and $\alpha = 0.1$ then we have:

$$\mu'_{l=-1}(x_a) = \alpha\left(\frac{\sum_{i=1}^{2} S\left(x_a, x_{b_i}\right)\left(\mu_{l=-1}\left(x_{b_i}\right) - \mu_{l=-1}(x_a)\right)}{2} - \frac{\sum_{i=1}^{5} D\left(x_a, x_{c_i}\right)\left(\mu_{l=-1}\left(x_{c_i}\right) - \mu_{l=-1}(x_a)\right)}{5} + \mu_{l=-1}(x_a)\right)$$

$+(1-\alpha)\mu_{l=-1}(x_a) = 0.18.$

As it can be seen the refined membership value of instance x_a to class -1 decreases from 0.5 to 0.18 for the given example. Since $y_a = +1$, the expected membership value of instance x_a in the positive class (EMV_-1)$\mu^E_{y_a=-1}(x_a) = 0$. Consequently the difference between RMV_-1 and EMV_-1 is less than that between unrefined membership value (UMV_-1) and EMV_-1:

$\left|\mu^E_{y_a=-1}(x_a) - \mu'_{l=-1}(x_a)\right| = 0.18 \leq \left|\mu^E_{y_a=-1}(x_a) - \mu_{l=+1}(x_a)\right| = 0.50.$

Theorem 4.2 demonstrates that the difference between the RMV_*y* and EMV_*y* of a given instance in the target domain is less than that between the UMV_*y* and EMV_*y*. Consequently, it proves that the error produced by the RMV_*y* is less than that gained by the UMV_*y*. In conclusion, Theorem 4.2 implies that the proposed refinement brings about more accurate prediction. Based on this theory the proposed algorithm is developed.

Theorem 4.2 Let $x_a \in D_t$, S and D are strongly (ε, γ) - good similarity and dissimilarity functions**,** $\mu_{l=y}(x)$, $\mu'_{l=y}(x)$ and $\mu^E_{l=y}(x)$ are UMV_*y*, RMV_*y* and EMV_y respectively. If $\frac{\beta}{\partial} > \frac{1-\theta}{\theta}$ then

A) $\mu'_{l=y_a}(x_a) \geq \mu_{l=y_a}(x_a)$ and

B)$\mu'_{l=-y_a}(x_a) \leq \mu_{l=-y_a}(x_a)$

and consequently $\forall\, y \in \{-y_a, y_a\}$,

$$\left|\mu^{E}_{l=y}(x_a) - \mu'_{l=y}(x_a)\right| \leq \left|\mu^{E}_{l=y}(x_a) - \mu_{l=y}(x_a)\right|,$$

where

$\mu^{E}_{l=y}(x_a) = 1$ if $y = y_a$ and $\mu^{E}_{l=y}(x_a) = 0$ if $y = -y_a$.

Proof: Part A)

$$\mu_{l=y_a}{}'(x_a) =$$

$$\alpha\left(\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y_a}\left(x_{b_i}\right) - \mu_{l=y_a}(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y_a}\left(x_{c_i}\right) - \mu_{l=y_a}(x_a)\right)}{K_\partial} + \mu_{l=y_a}(x_a)\right)$$

$$+(1-\alpha)\mu_{l=y_a}(x_a) =$$

$$\alpha\left(\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y_a}\left(x_{b_i}\right) - \mu_{l=y_a}(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y_a}\left(x_{c_i}\right) - \mu_{l=y_a}(x_a)\right)}{K_\partial}\right) + \mu_{l=y_a}(x_a).$$

Since $\alpha > 0$, if we show that

$$\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y_a}\left(x_{b_i}\right) - \mu_{l=y_a}(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y_a}\left(x_{c_i}\right) - \mu_{l=y_a}(x_a)\right)}{K_\partial} \geq 0,$$

or

$$\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y_a}\left(x_{b_i}\right) - \mu_{l=y_a}(x_a)\right)}{K_\beta} + \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y_a}(x_a) - \mu_{l=y_a}\left(x_{c_i}\right)\right)}{K_\partial} \geq 0,$$

then Part A will be proved:

From $S\left(x_a, x_{b_i}\right) \geq \beta$ and $\mu_{l=y_a}\left(x_{b_i}\right) \geq \theta$ we have:

$$S\left(x_a, x_{b_i}\right)\left(\mu_{l=y_a}\left(x_{b_i}\right) - \mu_{l=y_a}(x_a)\right) \geq \beta\left(\theta - \mu_{l=y_a}(x_a)\right). \tag{4.1}$$

From $\mu_{l=-y_a}\left(x_{c_i}\right) \geq \theta$ we have $\mu_{l=y_a}\left(x_{c_i}\right) \leq 1-\theta$ and then $-\mu_{l=y_a}\left(x_{c_i}\right) \geq \theta - 1$.

Therefore, from $D\left(x_a, x_{c_i}\right) \geq \partial$ we have:

$$D\left(x_a, x_{c_i}\right)\left(\mu_{l=y_a}(x_a) - \mu_{l=y_a}\left(x_{c_i}\right)\right) \geq \partial\left(\mu_{l=y_a}(x_a) + \theta - 1\right). \tag{4.2}$$

From Equations 4.1 and 4.2 we have:

$$\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y_a}\left(x_{b_i}\right) - \mu_{l=y_a}(x_a)\right)}{K_\beta} + \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y_a}(x_a) - \mu_{l=y_a}\left(x_{c_i}\right)\right)}{K_\partial}$$

$$\geq \frac{\sum_{i=1}^{K_\beta} \beta\left(\theta - \mu_{l=y_a}(x_a)\right)}{K_\beta} + \frac{\sum_{i=1}^{K_\partial} \partial\left(\mu_{l=y_a}(x_a) + \theta - 1\right)}{K_\partial}.$$

Since the terms $\beta\left(\theta-\mu_{l=y_a}(x_a)\right)$ and $\partial\left(\mu_{l=y_a}(x_a)+\theta-1\right)$ are constant values and independent from i, we have:

$$\frac{\sum_{i=1}^{K_\beta}\beta\left(\theta-\mu_{l=y_a}(x_a)\right)}{K_\beta}+\frac{\sum_{i=1}^{K_\partial}\partial\left(\mu_{l=y_a}(x_a)+\theta-1\right)}{K_\partial}$$

$$=\beta\left(\theta-\mu_{l=y_a}(x_a)\right)+\partial\left(\mu_{l=y_a}(x_a)+\theta-1\right)$$

$$=\beta\theta-\beta\mu_{l=y_a}(x_a)+\partial\mu_{l=y_a}(x_a)+\partial\theta-\partial$$

$$=\theta(\partial+\beta)+\mu_{l=y_a}(x_a)(\partial-\beta)-\partial.$$

If considering the worst case**:** $\mu_{l=y_a}(x_a)=0$ and from assumption $\frac{\beta}{\partial}>\frac{1-\theta}{\theta}$then we have:

$$\theta(\partial+\beta)+\mu_{l=y_a}(x_a)(\partial-\beta)-\partial$$

$$=\theta(\partial+\beta)-\partial>\theta\left(\partial+\left(\frac{\partial}{\theta}\right)(1-\theta)\right)-\partial$$

$$=\theta\partial+\partial(1-\theta)-\partial$$

$$=\theta\partial+\partial-\theta\partial-\partial=0\ \blacksquare$$

Proof: Part B)

$$\mu_{l=-y_a}{}'(x_a)=$$

$$\alpha\left(\frac{\sum_{i=1}^{K_\beta}S\left(x_a,x_{b_i}\right)\left(\mu_{l=-y_a}\left(x_{b_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\beta}-\frac{\sum_{i=1}^{K_\partial}D\left(x_a,x_{c_i}\right)\left(\mu_{l=-y_a}\left(x_{c_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\partial}+\mu_{l=-y_a}(x_a)\right)$$

$$+(1-\alpha)\mu_{l=-y_a}(x_a)=$$

$$\alpha\left(\frac{\sum_{i=1}^{K_\beta}S\left(x_a,x_{b_i}\right)\left(\mu_{l=-y_a}\left(x_{b_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\beta}-\frac{\sum_{i=1}^{K_\partial}D\left(x_a,x_{c_i}\right)\left(\mu_{l=-y_a}\left(x_{c_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\partial}\right)+\mu_{l=-y_a}(x_a).$$

Since $\alpha>0$, if we show that

$$\frac{\sum_{i=1}^{K_\beta}S\left(x_a,x_{b_i}\right)\left(\mu_{l=-y_a}\left(x_{b_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\beta}-\frac{\sum_{i=1}^{K_\partial}D\left(x_a,x_{c_i}\right)\left(\mu_{l=-y_a}\left(x_{c_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\partial}\leq 0,$$

or

$$\frac{\sum_{i=1}^{K_\partial}D\left(x_a,x_{c_i}\right)\left(\mu_{l=-y_a}\left(x_{c_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\partial}-\frac{\sum_{i=1}^{K_\beta}S\left(x_a,x_{b_i}\right)\left(\mu_{l=-y_a}\left(x_{b_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\beta}\geq 0,$$

or

$$\frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=-y_a}\left(x_{c_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\partial}+\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=-y_a}(x_a)-\mu_{l=-y_a}\left(x_{b_i}\right)\right)}{K_\beta} \geq 0,$$

then Part B will be proved:

From $D\left(x_a, x_{c_i}\right) \geq \partial$ and $\mu_{l=-y_a}\left(x_{c_i}\right) \geq \theta$ we have:

$$D\left(x_a, x_{c_i}\right)\left(\mu_{l=-y_a}\left(x_{c_i}\right)-\mu_{l=-y_a}(x_a)\right) \geq \partial\left(\theta-\mu_{l=-y_a}(x_a)\right). \quad (4.3)$$

From $\mu_{l=y_a}\left(x_{b_i}\right) \geq \theta$ we have $\mu_{l=-y_a}\left(x_{b_i}\right) \leq 1-\theta$ and then $-\mu_{l=-y_a}\left(x_{b_i}\right) \geq \theta-1$.

Therefore from $S\left(x_a, x_{b_i}\right) \geq \beta$ we have:

$$S\left(x_a, x_{b_i}\right)\left(\mu_{l=-y_a}(x_a)-\mu_{l=-y_a}\left(x_{b_i}\right)\right) \geq \beta\left(\mu_{l=-y_a}(x_a)+\theta-1\right). \quad (4.4)$$

From Equations 4.3 and 4.4 we have:

$$\frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=-y_a}\left(x_{c_i}\right)-\mu_{l=-y_a}(x_a)\right)}{K_\partial}+\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=-y_a}(x_a)-\mu_{l=-y_a}\left(x_{b_i}\right)\right)}{K_\beta}$$

$$\geq \frac{\sum_{i=1}^{K_\partial} \partial(\theta-\mu_{l=-y_a}(x_a))}{K_\partial}+\frac{\sum_{i=1}^{K_\beta} \beta\left(\mu_{l=-y_a}(x_a)+\theta-1\right)}{K_\beta}.$$

Since the terms $\partial(\theta-\mu_{l=-y_a}(x_a))$ and $\beta\left(\mu_{l=-y_a}(x_a)+\theta-1\right)$ are constant values and independent from i, we have:

$$\frac{\sum_{i=1}^{K_\partial} \partial\left(\theta-\mu_{l=-y_a}(x_a)\right)}{K_\partial}+\frac{\sum_{i=1}^{K_\beta} \beta\left(\mu_{l=-y_a}(x_a)+\theta-1\right)}{K_\beta}$$

$$=\partial\left(\theta-\mu_{l=-y_a}(x_a)\right)+\beta\left(\mu_{l=-y_a}(x_a)+\theta-1\right)$$

$$=\partial\theta-\partial\mu_{l=-y_a}(x_a)+\beta\mu_{l=-y_a}(x_a)+\beta\theta-\beta$$

$$=\theta(\partial+\beta)+\mu_{l=-y_a}(x_a)(\beta-\partial)-\beta.$$

If considering the worst case**:** $\mu_{l=-y_a}(x_a)=1$ and from assumption $\frac{\beta}{\partial} > \frac{1-\theta}{\theta}$then we have:

$$\theta(\partial+\beta)+(\beta-\partial)-\beta$$

$$=\theta(\partial+\beta)-\partial > \theta\left(\partial+\left(\frac{\partial}{\theta}\right)(1-\theta)\right)-\partial$$

$$=\theta\partial+\partial(1-\theta)-\partial$$

$$=\theta\partial+\partial-\theta\partial-\partial=0 \blacksquare$$

Theorem 4.2 proves that the RMV_y of instance x_a is closer to the EMV_y than UMV_y, so higher predictive accuracy will be achieved if the RMV_y is computed.

We define the RMV_y in iteration n by Definition 4.7. It then will be used in Theorem 4.3 to show that the proposed algorithm will converge to EMV_y.

Definition 4.7 Let $x_a \in D_t$, $\mu_{l=y}(x)$, $\mu'_{l=y}(x)$ and $\mu^E_{l=y}(x)$ are UMV_y, RMV_y and EMV_y respectively. $\forall y \in \{-1,\ 1\}$ we call $\mu^n_{l=y}(x_a)$ the RMV_y in step n with the following equation:

$$\mu^n_{l=y}(x_a) =$$

$$\alpha^n \left(\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y}\left(x_{b_i}\right) - \mu^{n-1}_{l=y}(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y}\left(x_{c_i}\right) - \mu^{n-1}_{l=y}(x_a)\right)}{K_\partial} + \mu^{n-1}_{l=y}(x_a) \right)$$

$$+(1-\alpha^n)\mu^{n-1}_{l=y}(x_a),$$

where $n = 0, \mu^0_{l=y}(x_a) = \mu_{l=y}(x_a)$ and $n = 1, \mu^1_{l=y}(x_a) = \mu'_{l=y}(x_a)$.

Finally, Theorem 4.3 proves that, if the refinement value is computed based on an iterative format, as proposed in the loop of Step 5 of the algorithm, it will converge and reach to EMV_y.

Theorem 4.3 Let $x_a \in D_t$, $\mu^E_{l=y}(x_a)$ and $\mu^n_{l=y}(x_a)$ are EMV_y and RMV_y respectively. If $\alpha^n < \frac{1-\mu^{n-1}_{l=y}(x_a)}{\gamma^{n-1}}$ then

$$\forall y \in \{-1, +1\}, \lim_{n\to\infty} \mu^n_{l=y}(x_a) = \mu^E_{l=y}(x_a) = \begin{cases} 1 & y = y_a \\ 0 & y = -y_a \end{cases},$$

where $\gamma^n = \frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y}\left(x_{b_i}\right) - \mu^{n-1}_{l=y}(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y}\left(x_{c_i}\right) - \mu^{n-1}_{l=y}(x_a)\right)}{K_\partial}$.

Proof: To prove the theorem it is enough to prove that the RMV_y is bounded and monotonic.

Part A) $\lim_{n\to\infty} \mu^n_{l=y_a}(x_a) = \mu^E_{l=y_a}(x_a) = 1$.

It is concluded that the $\mu^n_{l=y}(x_a)$ is increasing from the Theorem 4.2. So we just need to prove that it is bounded. Since it is nonnegative value, we need to prove that:

$0 \le \mu^n_{l=y}(x_a) \le 1$

So we have:

$\mu_{l=y}^{n}(x_a)$

$$= \alpha^n \left(\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y}\left(x_{b_i}\right) - \mu_{l=y}^{n-1}(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y}\left(x_{c_i}\right) - \mu_{l=y}^{n-1}(x_a)\right)}{K_\partial} + \mu_{l=y}^{n-1}(x_a) \right)$$

$$+(1-\alpha^n)\mu_{l=y}^{n-1}(x_a)$$

$$= \alpha^n \left(\frac{\sum_{i=1}^{K_\beta} S\left(x_a, x_{b_i}\right)\left(\mu_{l=y}\left(x_{b_i}\right) - \mu_{l=y}^{n-1}(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} D\left(x_a, x_{c_i}\right)\left(\mu_{l=y}\left(x_{c_i}\right) - \mu_{l=y}^{n-1}(x_a)\right)}{K_\partial} \right) + \mu_{l=y}^{n-1}(x_a)$$

$$= \alpha^n(\gamma^{n-1}) + \mu_{l=y}^{n-1}(x_a) < \frac{1-\mu_{l=y}^{n-1}(x_a)}{\gamma^{n-1}}\gamma^{n-1} + \mu_{l=y}^{n-1}(x_a) = 1 \blacksquare$$

Part B) $\lim_{n\to\infty} \mu_{l=-y_a}^{n}(x_a) = \mu_{l=-y_a}^{E}(x_a) = 0$.

$\mu_{l=-y}^{n}(x_a)$ is nonnegative and decreasing and the proof is similar to the Part A ■

4.3.2 Multi-Step Fuzzy Bridged Refinement-based Algorithm

Given $\tilde{F}^s = \{\tilde{f}_1^s, \dots, \tilde{f}_m^s\}$ and $\tilde{F}^t = \{\tilde{f}_1^t, \dots, \tilde{f}_m^t\}$ are the fuzzy feature sets for source domain D_s and target domain D_t respectively, where $\tilde{f}_l$ is a fuzzy membership function for each feature. DIC, which is a novel self organizing clustering technique, is applied to create the fuzzy features. DIC is a dynamic clustering technique avoiding drawbacks such as stability-plasticity and *inflexibility* found in other methods and computing trapezoidal-shaped fuzzy sets (Tung et al. 2004). It is assumed that the number of these features for both target and source domains is the same, but the membership functions of these fuzzy sets are different. This assumption implies a need for transductive transfer learning in which the feature space is the same, but the distributions are different. Given $X^s = \left\{x_1^s, \dots, x_{n_s}^s\right\}$ are source domain instances and, $X^{t_0} = \left\{x_1^{t_0}, \dots, x_{n_{t_0}}^{t_0}\right\}$ *and* $X^{t_1} = \left\{x_1^{t_1}, \dots, x_{n_{t_1}}^{t_1}\right\}$ are the instances of target domain with labels and without labels respectively, where $n_{t_0} \ll n_{t_1}$ where $\ll$ means n_{t_0} much less than n_{t_1}. Given $\tilde{Y} = \{\tilde{y}_1, \dots, \tilde{y}_l\}$ is the predictive fuzzy label set, which is the same for both domains. Given $G(\cdot)$ is a shift-unaware predictive model, which is a fuzzy neural network (Behbood et al. 2010) in this paper, so that $G(x_i) =$

$\{\mu_{\tilde{y}_1}(x_i), \dots, \mu_{\tilde{y}_l}(x_i)\}$ is the vector of membership values of x_i that belongs to each label. Given $sm(\cdot)$ and $dm(\cdot)$ are strongly (ε, γ) - good similar and dissimilar functions defined in Definition 3.2. The Fuzzy Bridged Refinement (FBR) algorithm is described as follows:

<Fuzzy Bridged Refinement algorithm>

Input: *Source domain:* D_s

Target domain: D_t

Fuzzy feature space of target domain: $\tilde{F}^t$

Fuzzy feature space of source domain: $\tilde{F}^s$

Predictive fuzzy labels: $\tilde{Y}$

Prediction model: $G(\cdot)$

Similarity function: $sm(\cdot)$

Dissimilarity function: $dm(\cdot)$

Coefficient parameter: α

Output: *A label matrix* $\widetilde{MR} = (\widetilde{MR}_{ij})_{n_{t_1} \times l}$ *for unlabeled instances of the target domain* X^{t_1}.

[Begin]

Step 1: The Singleton fuzzifier is used as follows to fuzzify the crisp-value of instances from both domains.

$$\mu_{\tilde{X}_i}(\tilde{x}_i) = \begin{cases} 1, & \text{if} \quad \tilde{x}_i(T) = x_i(T) \\ 0, & \text{if} \quad Otherwise \end{cases},$$

where $\tilde{x}_i$ is the fuzzified equivalent of crisp input x_i , $i \in \{1,2, \dots, n\}$.

Step 2: To perform antecedent matching of fuzzyfied inputs x_i against fuzzy features $\tilde{F}^s$ and $\widetilde{F}^t$, the input membership value in each feature is computed as follows:

$$\mu_{\tilde{f}_j}(\tilde{x}_i) = \begin{cases} 0, & \text{if} \quad x_i \le l_{\tilde{f}_j} \\ \frac{x_i - l_{\tilde{f}_j}}{u_{\tilde{f}_j} - l_{\tilde{f}_j}}, & \text{if} \quad l_{\tilde{f}_j} \le x_i \le u_{\tilde{f}_j} \\ 1, & \text{if} \quad u_{\tilde{f}_j} \le x_i \le v_{\tilde{f}_j} \\ \frac{r_{\tilde{f}_j} - x_i}{r_{\tilde{f}_j} - u_{\tilde{f}_j}}, & \text{if} \quad v_{\tilde{f}_j} \le x_i \le r_{\tilde{f}_j} \\ 0, & \text{if} \quad x_i \ge r_{\tilde{f}_j} \end{cases} \quad i \in \{1,2,\dots,n\},\ j \in \{1,2,\dots,m\},$$

Step 3: The fuzzy initial label matrix for unlabeled target domain instances ($G^{t_1}_{n_{t_1}\times l}$) is calculated by the shift-unaware fuzzy neural network, which is trained by source domain data and the fuzzy label matrix for source domain instances ($G^{s}_{n_s\times l}$). The fuzzy label matrix for labeled target instances ($G^{t_0}_{n_{t_0}\times l}$) are considered as follows:

$$G^{t_1}_{n_{t_1}\times l} = \{g^{t_1}_{(i,j)} |\ g^{t_1}_{(i,j)} = FNN(x_i^{t_1}) = \left(\mu_{\widetilde{y_j}}(x_i^{t_1})\right\}.$$

$$G^{s}_{n_s\times l} = \left\{g^{s}_{(i,j)} | g^{s}_{(i,j)} = \mu_{\widetilde{y_j}}(x_i^{s})\right\}.$$

$$G^{t_0}_{n_{t_0}\times l} = \{g^{t_0}_{(i,j)} | g^{t_0}_{(i,j)} = \mu_{\widetilde{y_j}}(x_i^{t_0})\}.$$

Step 4: The similarity and dissimilarity matrixes of the instances in domain D and the unlabeled instances in the target domain is calculated:

$SM = \{sm(i,j), i = 1 \dots |D|, j = 1 \dots n_{t_1}\}$,

$DM = \{dm(i,j), i = 1 \dots |D|, j = 1 \dots n_{t_1}\}$,

where D is the mixture of source and target domains and specified as input in each setting that will be described in the next section. $|D|$ is the number of instances in D.

Step 5: For given $\beta, \partial > 0$ the sets of most similar $KSM\_\beta(x_i)$ and most dissimilar $KDM\_\partial(x_i)$ instances to each unlabeled target instance $x_i \in X^{t_1}$ are computed:

For $i = 1$ to n_{t_1}

$KSM\_\beta_{(x_i)} = \{n_1^i, \dots, n_{k_\beta}^i | n_p^i \in D\}$

$KDM\_\partial_{(x_i)} = \{v_1^i, \dots, v_{k_\partial}^i | v_p^i \in D\}$

Next i

Step 6: The initial fuzzy label for each unlabeled instance from the target domain is refined in this step. $\widetilde{MR}^{w}_{n_{t_1}\times l}$ is the fuzzy label matrix for instances of the target domain in step *w* in the Multi-Step Fuzzy Bridged Refinement-based algorithm.

Repeat

For $i = 1\ to\ n_{t_1}$

For $j = 1\ to\ l$

$$\widetilde{MR}(i,j)^{w}_{t} = \propto_{t-1}\left(\frac{(\sum_{p=1}^{k_\beta} sm(i,n_p^i)(\mu_{\tilde{y}_j}(n_p^i)-\widetilde{MR}^{w}_{t-1}(i,j))}{k_\beta} - \frac{(\sum_{p=1}^{k_\partial} dm(i,v_p^i)(\mu_{\tilde{y}_j}(v_p^i)-\widetilde{MR}^{w}_{t-1}(i,j))}{k_\partial} + \widetilde{MR}(i,j)^{w}_{t-1}\right) + (1-\propto_{t-1}) g(i,j)^{t_1}$$

Next j

Next i

For $j = 1$ to l

$$\widetilde{MR}(i,j)^{w}_{t} = \widetilde{MR}(i,j)^{w}_{t} / (\frac{S_j}{r^j n_{t_1}})$$

Next j

$t = t + 1$

Until $\widetilde{MR}$ converges

[End]

As can be seen, the refinement is based on the fact that the label of the most similar and the least similar instances to the target instance is used to modify the initial label of target instance, which was initialized by a trained shift-unaware fuzzy neural network. As the result of the FBR algorithm a fuzzy label matrix for all unlabeled instances of the target domain, $(\widetilde{MR}_{ij})_{n_{t_1}\times l}$, is achieved. Each row of this matrix indicates the membership values of one instance in all label classes.

Moreover, it assumes that each instance belongs to a specified label if it gets the maximum membership value for this label between other label classes. Given the value $S_j = \sum_{i=1}^{n_{t_1}} Label(x_i^{t_1})$ indicates the number of samples belonging to label class $j = 1, \dots, l$. Although it is changing during the iteration, it is expected to reach the actual number of instances belonging to label class j, and may bring about poor

performance, particularly for imbalanced data. Considering an imbalanced problem, given r^j is the ratio of instances in class label j. To solve the problem, the class ratio normalization is applied. In each iteration, the class ratio is normalized to the true class ratio that is provided. The algorithm is able to reduce the influence of the imbalanced data set on the predictive accuracy using normalization. The imbalanced problems are widespread in real world applications where the target class, which needs to be recognized and predicted, has fewer instances than other classes.

To find the final label for each instance ($x_i^{t_1}$), following equation can be used as defuzzifier.

Label $(x_i^{t_1}) = \arg max_j \{\widetilde{MR}_{i,j} | j = 1, \dots, l\}$.

The Fuzzy Bridged Refinement-based algorithm can be applied at least in the two-step refinement process, which firstly refines the labels towards $D = X^s \cup X^{t_1}$, and then toward $D = X^{t_1}$. The results of the two-step FBR algorithm (2SFBR) have demonstrated significant improvement in comparison with shift-unaware classifiers. However, the accuracy of each data set follows the performance of the shift-unaware classifier and, consequently, has poor performance in some cases, which will be described in the Experiments section. Also, it doesn't take the impact of existing labeled data in the target domain into account. To solve these problems and improve the predictive accuracy, we propose to have multiple steps to refine the initial labels and assume the existence of a few labeled instances in the target domain. The refinement process moves from source domain (D_s) toward target domains $(D_{t_1} or\ D_{t_0})$ through n steps using the trade-off parameter μ, which indicates the percentage of instances of the source domain and target domain in the mixture domain in each step of refinement. As n increases. μ becomes greater, and consequently, the contribution of source domain data in the mixture domain becomes less and conversely, the portion of target domain data increases. Accordingly, the consequent neighboring mixture domains are similar to each other and smoothly transfer from the source domain toward the target domain. Through the multi-step process, it is able to make a bridge and transfer the label structure between the source

and target domains more accurately and easily. Figure 4.1 demonstrates the proposed Multi-Step Fuzzy Bridged Refinement-based (MSFBR) algorithm.

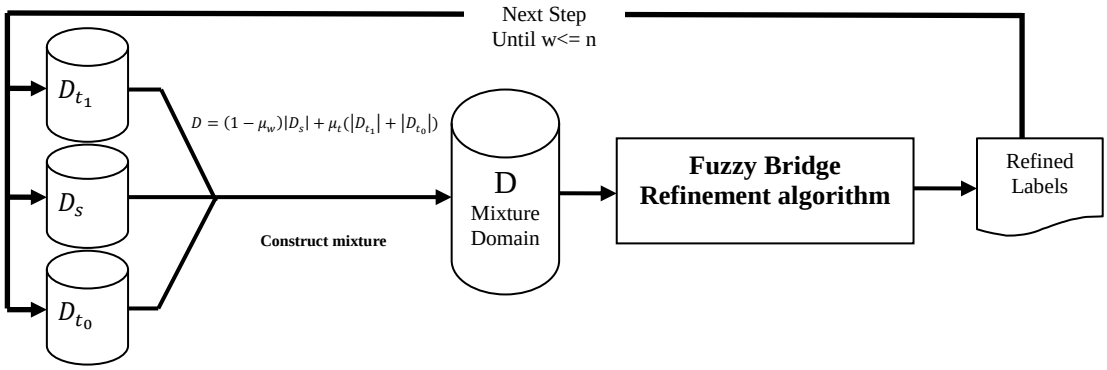


FIGURE 4.1: MULTI-STEP FUZZY BRIDGED REFINEMENT-BASED DOMAIN ADAPTATION

The MSFBR for Transductive Transfer Learning algorithm is described as follows:

<Multi-Step Fuzzy Bridged Refinement-based algorithm>

Input: *Source domain:* D_s

Target domain: D_t

Fuzzy feature space of target domain: $\tilde{F}^t$

Fuzzy feature space of source domain: $\tilde{F}^s$

Predictive fuzzy labels: $\tilde{Y}$

Prediction model: $G(\cdot)$

Similarity function: $sm(\cdot)$

Dissimilarity function: $dm(\cdot)$

Coefficient parameter: α

Output: *Label* $(x_i^{t_1}) = arg\, max_j \{\widetilde{MR}_{i,j}^n | j = 1, \dots, l\}$

[Begin]

Step 1:

$$\mu_{\tilde{X}_i}(\tilde{x}_i) = \begin{cases} 1, & \text{if } \tilde{x}_i(T) = x_i(T), \\ 0, & \text{if } Otherwise \end{cases} \qquad i \in \{1,2,\dots,n\}.$$

Step 2:

$$\mu_{\tilde{f}_j}(\tilde{x}_i) = \begin{cases} 0, & \text{if} \quad x_i \le l_{\tilde{f}_j} \\ \frac{x_i - l_{\tilde{f}_j}}{u_{\tilde{f}_j} - l_{\tilde{f}_j}}, & \text{if} \quad l_{\tilde{f}_j} \le x_i \le u_{\tilde{f}_j} \\ 1, & \text{if} \quad u_{\tilde{f}_j} \le x_i \le v_{\tilde{f}_j} \\ \frac{r_{\tilde{f}_j} - x_i}{r_{\tilde{f}_j} - u_{\tilde{f}_j}}, & \text{if} \quad v_{\tilde{f}_j} \le x_i \le r_{\tilde{f}_j} \\ 0, & \text{if} \quad x_i \ge r_{\tilde{f}_j} \end{cases} \quad i \in \{1,2,\dots,n\},\ j \in \{1,2,\dots,m\}.$$

Step 3:

$$G^{t_1}_{n_{t_1}\times l} = \left\{ g^{t_1}_{(i,j)} | g^{t_1}_{(i,j)} = FNN\left(x_i^{t_1}\right) = \mu_{\tilde{y}_j}\left(x_i^{t_1}\right) \right\};$$

$$G^{s}_{n_s\times l} = \left\{ g^{s}_{(i,j)} | g^{s}_{(i,j)} = \mu_{\tilde{y}_j}(x_i^{s}) \right\};$$

$$G^{t_0}_{n_{t_0}\times l} = \left\{ g^{t_0}_{(i,j)} \middle| g^{t_0}_{(i,j)} = \mu_{\tilde{y}_j}\left(x_i^{t_0}\right) \right\};$$

Step 4:

For $w = 1\ to\ n$

$\mu_w = w/n$

$D = (1-\mu_w)|D_s| + \mu_w(|D_{t_1}| + |D_{t_0}|)$

Step 4-1:

$SM = \{sm(i,j), i = 1 \dots |D|, j = 1 \dots\ n_{t_1}\}$

$DM = \{dm(i,j), i = 1 \dots |D|, j = 1 \dots\ n_{t_1}\}$

Step 4-2:

For $i = 1$ to n_{t_1}

$KSM\_\beta_{(x_i)} = \{n_1^i, \dots, n_{k_\beta}^i | n_p^i \in D\}$

$KDM\_\partial_{(x_i)} = \{v_1^i, \dots, v_{k_\partial}^i | v_p^i \in D\}$

Next i

Step 4-3:

Do
For $i = 1$ to n_{t_1}

For $j = 1$ to l

$\widetilde{MR}(i,j)^w_{t-1}$

$$= \propto_{t-1} \left(\frac{(\sum_{p=1}^{k_\beta} sm(i,n_p^i)(\mu_{\tilde{y}_j}(n_p^i) - \widetilde{MR}_{t-1}^w(i,j))}{k_\beta} - \frac{(\sum_{p=1}^{k_\partial} dm(i,v_p^i)(\mu_{\tilde{y}_j}(v_p^i) - \widetilde{MR}_{t-1}^w(i,j))}{k_\partial} + \widetilde{MR}(i,j)_{t-1}^w \right)$$

$$+(1 - \propto_{t-1}) g(i,j)^{t_1}$$

Next j

Next i

For $j = 1$ to l

$$\widetilde{MR}(i,j)_t^w = \widetilde{MR}(i,j)_t^w / (\frac{S_j}{r^j n_{t_1}})$$

Next j

$t = t + 1$

Until *MR* converges

Next w

[End]

To find out the best performance of the MSFBR algorithm it is examined and analyzed using two different settings: Setting 1, which ignores the few labeled instances in the target domain (X^{t_0}); Setting 2, which takes these labeled instances into account. They can be depicted as follows:

Setting 1: Call MSFBR algorithm where $D_{t_0} = \emptyset$.

Setting 2: Call MSFBR algorithm where $D_{t_0} \neq \emptyset$.

It should be mentioned that the above algorithm is implemented by Matlab, which has the ability of matrix language programming. The performance of the proposed MSFBR algorithm using these settings is explained in Section 4.4.

4.4 Experiments and Empirical Analysis

In this section we present a set of experiments to validate the proposed MSFBR algorithm using real-world bank failure data in which the prediction label has two classes: Failure; and, Survived. We perform the experiments to examine the MSFBR algorithm's performance to transfer a label structure from different time periods, which improves a long-term prediction capability for a FNN. The predictive accuracy of the proposed MSFBR algorithm is examined using eight different settings.

Likewise, the performance of the MSFBR algorithm to refine the predicted labels resulting from the fuzzy neural network as a shift-unaware predictor is compared with the performance of three famous methods as the baseline predictors. They are Transductive Support Vector Machine (TSVM), Naïve Bayes and Support Vector Machine (SVM) using 2SFBR algorithm (Xing et al. 2007). The results demonstrate a significant improvement which is proved by statistical tests.

4.4.1 DATA SETS

The data sets and financial variables used in the experiments are extracted from Call Report Data, which is downloaded from the website of the Federal Reserve Bank of Chicago[8] and the status of each bank is identified according to the Federal Financial Institutions Examination Council (FFIEC)[9] . The data set, which is shown in Table 4.3, includes the observation period of the survived banks of 21 years from Jun 1980 to Dec 2000, based on the history of each bank in FFIEC. There are 548 failed banks and 2555 survived ones. Although Tung et al. (2004) used nine financial features according to their statistical significance and correlation, it is observed that the model with three features has less created rules, less computational load and greater prediction accuracy. Each feature is ranked based on the importance of a feature as a result of a future selection process and three features with the highest grade are selected (Ng et al. 2008). The definitions of all features are described in Table 2.1. The experiments are run by nine, and three, features separately and the results are then compared. These features are widely accepted as being the most significant features in the bank failure literature. The domain instances X^s are selected from the data set until year 1990. The data set is used as training data. The target instances X^{t_1} (test data) and $X^{t_0}$are selected from records of years 1995, 1998 and 2000, 5, 8 and 10 years respectively after 1990 respectively.

[8] **http://www.chicagofed.org**
[9] **http://www.ffiec.gov/nicpubweb/nicweb/NicHome.aspx**

TABLE 4.3: BANK RECORDS IN DATA SET

Year	Total Number of banks	Number of survived banks	Number of failed banks
1990	2156	1843(85.48%)	313 (14.52%)
1995	2539	2192(86.34%)	347 (13.66%)
1998	2943	2585(87.84%)	358 (12.16%)
2000	3103	2555(82.34%)	548 (17.66%)

4.4.2 RESEARCH DESIGN AND COMPARISON

To reduce the influence of the imbalanced data-sets problem, the SMOTE (Chawla et al. 2002) is applied to the training data set. The number of failed banks increases to the number of survived banks to achieve a balanced data set, which improves the accuracy of prediction without losing important information. In each experiment, the training data set splits into two pools: (1) failed banks denoted with output "1"; (2) survived banks denoted with output "0". The 5-fold cross validation method is applied for training. The predictors are trained using training data sets and then evaluated by the testing data sets. The accuracy of the experiment in each scenario, which is the mean accuracy of cross-validation groups, is measured and calculated using GM. To specify the similarity and dissimilarity functions and construct the classifier, we follow the approach introduced in (Wang et al. 2008). The proposed approach, called *DBoost,* is applied to find the labels of similar and dissimilar instances using the Euclidean distance. The DBoost algorithm first constructs the pairs of positive and negative samples by considering all possible pairs of examples with different labels in data set. Then it is served as the training set for Boosting to learn the final large-margin convex-combination classifier. In the DBoost algorithm, AdaBoost (Freund & Schapire 1996) is selected as the booster due to its good ability to generate large-margin classifier.

Firstly, the accuracy of the refined results of fuzzy neural networks using two settings of MSFBR and 2SFBR with and without class ratio normalization are compared to evaluate the performance of the proposed algorithm. These comparisons are performed to find out the improvement gained due to the MSFBR algorithm and to find the best setting for the proposed algorithm. In 2SFBR, two refinements are carried out by applying $\mu_1 = 0.5$ and $\mu_2 = 1$. Likewise, the evaluation is performed using two categories of feature sets: nine variables and three variables, respectively. In conclusion, there are 16 experiments in this phase. These experiments and their denotations are shown in Table 4.4.

TABLE 4.4: DIFFERENT SETTINGS OF PROPOSED ALGORITHM

	Number of features	**Number of Steps**	**Class ratio normalization**	**Setting**
FNN_2SFBR-1	3	2	No	1
FNN_2SFBR-2	9	2	No	1
FNN_2SFBR-3	3	2	Yes	1
FNN_2SFBR-4	9	2	Yes	1
FNN_2SFBR-5	3	2	No	2
FNN_2SFBR-6	9	2	No	2
FNN_2SFBR-7	3	2	Yes	2
FNN_2SFBR-8	9	2	Yes	2
FNN_MSFBR-1	3	22	No	1
FNN_MSFBR-2	9	22	No	1
FNN_MSFBR-3	3	22	Yes	1
FNN_MSFBR-4	9	22	Yes	1
FNN_MSFBR-5	3	22	No	2
FNN_MSFBR-6	9	22	No	2
FNN_MSFBR-7	3	22	Yes	2
FNN_MSFBR-8	9	22	Yes	2

Secondly the possible improvement gained using fuzzy approach is investigated. The Multi Layer Perception (MLP) networks (Lin & Lee 1996) are trained as unaware-shift prediction models using the crisp-value financial features. They have 9-10-1 and 3-5-1 structures that have been empirically determined to provide optimal results for scenarios with nine covariates and scenarios with three variables, respectively. The best setting of the non-fuzzy version of the proposed algorithm (MSFBR) is then applied to refine the computed labels. The results are compared with those of the best setting of MSFBR refining the labels calculated by fuzzy neural network.

Finally, the performance of the best setting of the proposed algorithm on labels predicted by a fuzzy neural network is compared with the performance of the two-step bridged refinement algorithm (2SBR) (Xing et al. 2007) on labels predicted by famous predictors including: (1) Support Vector Machine (SVM) (Joachims 1999a) which is a powerful supervised learning algorithm. In the experiments SVM with linear kernel is used and all options set by default; (2) Naïve Bayes (Caruana & Niculescu-Mizil 2006) which performs remarkably well much of the research, despite its simplicity; and (3) Transductive Support Vector Machine (TSVM) (Joachims 1999b) which is a state-of-the-art semi-supervised learning algorithm.

4.4.3 EXPERIMENT RESULTS ANALYSIS

This section reports the results gained from the experiments. The comparisons are examined by statistical tests to ensure that the MSFBR algorithm achieves a significant improvement.

4.4.3.1 RESULTS ANALYSIS USING DIFFERENT SETTINGS

As previously mentioned, the algorithm is evaluated according to various settings to explore the best performance of MSFBR. These settings consider different situations ranging from using source data and unlabeled target data in two steps, to utilizing a few labeled target data as well in multiple steps. Applying different settings assists in finding out how different steps and the labeled target data can influence algorithm performance. Figure 4.2 depicts the accuracy of the proposed algorithm with different settings for three time periods of prediction while the FNN is applied as a predictor. The proposed algorithm has three parameters: K , n and α, which need to be set to perform experiments. We set K , n and α 70, 22 and 0.65 respectively in the experiments, which will be discussed in the next section.

As the results of these experiments, Figure 4.2 represents the accuracy of the proposed algorithm using different settings. Clearly, the proposed algorithm outperforms the fuzzy neural network in all settings. The best relative increase, which is gained in FNN_MSFBR-8, is achieved by 19%, 23% and 25% on 1995, 1998 and

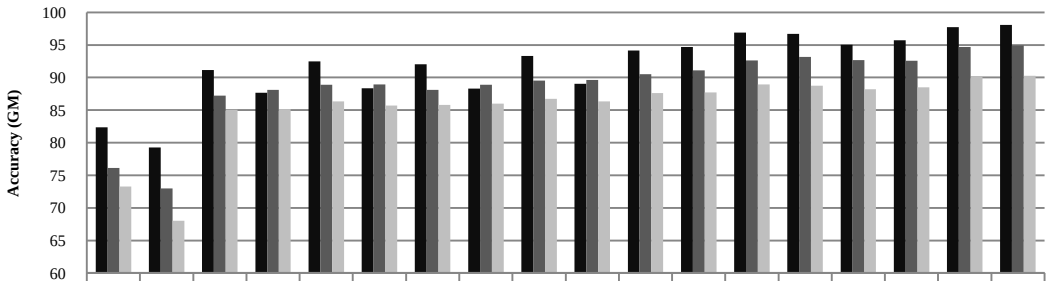
2000 data sets respectively where nine features are utilized. Surprisingly, this shows that the influence of the proposed algorithm becomes more significant once the period of prediction becomes more distant, and so the difference between the target domain and the source domain becomes greater. This growth in accuracy, which is gained by applying the proposed algorithm, is due to an experiment with nine features being used, in comparison with those with three variables. The reason may be that the proposed algorithm works better on larger feature spaces and with larger numbers of features.

Three statistical t-tests are carried out based on the following three criteria to analyse the performance of these settings and evaluate the influence of the proposed algorithm extension:

(1) The influence of the number of steps: whether it makes significant improvement or not. As can be seen from Figure 4.2 all settings of the FNN_MSFBR algorithm have better performance than FNN_2SFBR. To justify this improvement, the t-test is applied to compare the results of the corresponding settings on three data sets considering a level of significance $\alpha = 0.01$;

(2) The impact of the class ratio normalization on the accuracy. Figure 4.2 demonstrates that class ratio normalization enhances the accuracy. To further clarify this enhancement the t-test in level of significance $\alpha = 0.01$ is applied to compare the corresponding settings on all three data sets;

(3) The effect of applying labeled target data in the refinement algorithm. It can be implied from Figure 4.2 that employing labeled target instances reduces predictive error. To test this improvement, the t-test is performed on corresponding settings for all data sets in level of significance $\alpha = 0.01$.



	FNN-3	FNN-9	FNN_2SFBR-1	FNN_2SFBR-2	FNN_2SFBR-3	FNN_2SFBR-4	FNN_2SFBR-5	FNN_2SFBR-6	FNN_2SFBR-7	FNN_2SFBR-8	FNN_MSFBR-1	FNN_MSFBR-2	FNN_MSFBR-3	FNN_MSFBR-4	FNN_MSFBR-5	FNN_MSFBR-6	FNN_MSFBR-7	FNN_MSFBR-8
1995	82.36	79.25	91.15	87.66	92.45	88.34	92.02	88.31	93.29	89.04	94.15	94.66	96.89	96.71	95.03	95.72	97.74	98.07
1998	76.15	72.98	87.22	88.1	88.87	88.93	88.11	88.88	89.54	89.61	90.51	91.12	92.62	93.18	92.67	92.56	94.66	94.95
2000	73.29	68.03	85.07	85.17	86.32	85.69	85.78	86.01	86.75	86.35	87.63	87.71	88.94	88.76	88.22	88.49	90.15	90.26

FIGURE 4.2: ACCURACY OF PROPOSED ALGORITHM USING 16 EXPERIMENTS FOR 5, 8 AND 10 YEARS AHEAD PREDICTION

The results of these statistical tests are addressed in Table 4.5. As can be seen, all null hypothesizes of the equality of mean accuracy are rejected in 99% of confidence because the *P*-value is 0 and less than 0.01 for all hypotheses. It is concluded that these three issues have brought about significant performance improvement.

TABLE 4.5: T-TEST RESULTS TO EXAMINE DIFFERENT SETTINGS OF PROPOSED ALGORITHM

Setting	Mean Value	Hypothesis	t	DF	p-value	Result
FNN_2SFBR	88.28	FNN_2SFBR vs. FNN_MSFR	-11.063	23	0.00	Rejected
FNN_MSFR	92.56					
With Normalization	91.17	With Normalization vs. Without Normalization	10.523	23	0.00	Rejected
Without Normalization	89.66					
$D_{t_0} \neq \emptyset$	89.91	$D_{t_0} \neq \emptyset$ vs. $D_{t_0} \neq \emptyset$	-10.701	23	0.00	Rejected
$D_{t_0} = \emptyset$	90.93					

4.4.3.2 RESULTS ANALYSIS COMPARING MSFBR AND MSBR

This section addresses the benchmark of the best setting of the MSFBR and MSBR when $D_{t_0} \neq \emptyset$ and $D_{t_0} = \emptyset$. These algorithms are performed on twelve different case studies to investigate the influence of fuzzy approach of the proposed algorithm. The unrefined labels are computed using MLP networks and then are refined by MSBR. The results are presented in Table 4.6. As it can be seen the FNN_MSFBR outperforms the NN_MSBR in almost all scenarios.

TABLE 4.6: THE ACCURACY OF MSFRB AND MSBR ALGORITHMS

Labeled Target Data	Features	Year of Prediction	FNN_MSFBR	NN_MSBR
		1995	96.89	95.25
$D_{t_0} = \emptyset$	3	1998	92.62	90.33
		2000	88.94	86.07
	9	1995	96.71	95.74
		1998	93.18	93.56
		2000	88.76	85.81
$D_{t_0} \neq \emptyset$	3	1995	97.74	95.32
		1998	94.66	91.69
		2000	90.15	88.57
	9	1995	98.07	96.48
		1998	94.95	92.14
		2000	90.26	87.39

To testify the growth in accuracy, the Holm test (Holm 1979), which is a non-parametric statistical test, is applied to specify whether the improvement is significant or not. The result of the statistical test, which is presented in Table 4.7, rejects the equality of the accuracy in level of 0.05 of confidence and demonstrates that FFN_MSFBR outperforms NN_MSBR significantly. This comparison implies that the fuzzy approach played a significant role in the improvement in predictive accuracy.

TABLE 4.7: HOLM TEST FOR COMPARISON OF MSFBR AND MSFBR

Hypothesis	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Result
FNN_MSFBR vs. NN_MSBR	2.887	0.004	0.05	Rejected

4.4.3.3 RESULTS ANALYSIS USING DIFFERENT PREDICTION METHODS

In this section, the best corresponding settings of the proposed algorithm are compared with three prediction methods including SVM, Naïve Bayes and TSVM, which apply 2SBR. They are denoted as SVM_2SBR, NB_2SBR and TSVM_2SBR $D_{t_0} \neq \emptyset$ and $D_{t_0} = \emptyset$. These algorithms are benchmarked for twelve cases of study. These experiments and their results are summarized in Table 4.8. According to this table, the proposed algorithm outperforms other methods in all experiments in data sets of 1995, 1998 and 2000. To justify this improvement the Holm test in the level of

significance of $\alpha = 0.05$ is applied to show the difference in the performance of the algorithms.

Table 4.9, which is associated with the Holm procedure, shows all computations. The algorithms are ordered with respect to the z-value obtained. The normal distribution is applied to gain the corresponding p-value associated with each comparison. It is then compared with the associated α-Holm in the same row of the table to show whether the corresponding hypothesis of equal mean accuracy is rejected in favour of the FNN_MSFBR algorithm or not. The tests reject all hypotheses of equity of mean accuracy. Accordingly, it can be concluded that the FNN_MSFBR significantly reduces long-term predictive error and enhances the transfer of label structure from source domain to target domain.

4.4.4 PARAMETER SENSITIVITY

The proposed MSFBR algorithm has three parameters K, n and α which need to be set in performing experiments. In this section we empirically investigate the influence of these parameters on the performance of the proposed algorithm. To do this, the average accuracy of the algorithm on three data sets is examined using different values of these parameters on four settings of the proposed algorithm with settings MSFBR-2, 4, 6 and 8.

TABLE 4.8: THE ACCURACY OF BENCHMARKED ALGORITHMS

Labeled Target Data	Features	Year of Prediction	FNN_MSFBR	TSVM_2SBR	SVM_2SBR	NB_2SBR
$D_{t_0} = \emptyset$	3	1995	96.89	94.12	93.54	92.76
		1998	92.62	89.18	88.64	88.23
		2000	88.94	85.41	85.03	82.21
	9	1995	96.71	93.84	92.15	91.04
		1998	93.18	90.22	89.57	89.26
		2000	88.76	84.73	83.56	83.33
$D_{t_0} \neq \emptyset$	3	1995	97.74	94.15	93.22	93.45
		1998	94.66	92.21	92.69	90.93
		2000	90.15	89.29	88.15	87.38
	9	1995	98.07	95.37	94.26	94.11
		1998	94.95	91.52	91.14	89.32
		2000	90.26	86.79	85.97	85.56

TABLE 4.9: HOLM TEST FOR COMPARISON OF PROPOSED ALGORITHM WITH TSVM_2BR, SVM_2BR AND NB_2BR

Algorithms	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Hypothesis
NB_2BR	5.534	3.130E-8	0.017	Rejected
SVM_2BR	3.795	1.478E-4	0.025	Rejected
TSVM_2BR	2.055	0.039	0.05	Rejected

The accuracy of the algorithm for different values of K is shown in Figure 4.3. It shows that the performance is not greatly sensitive to K as long as K is large enough and the value of 80 is the best value for K that is chosen in this research.

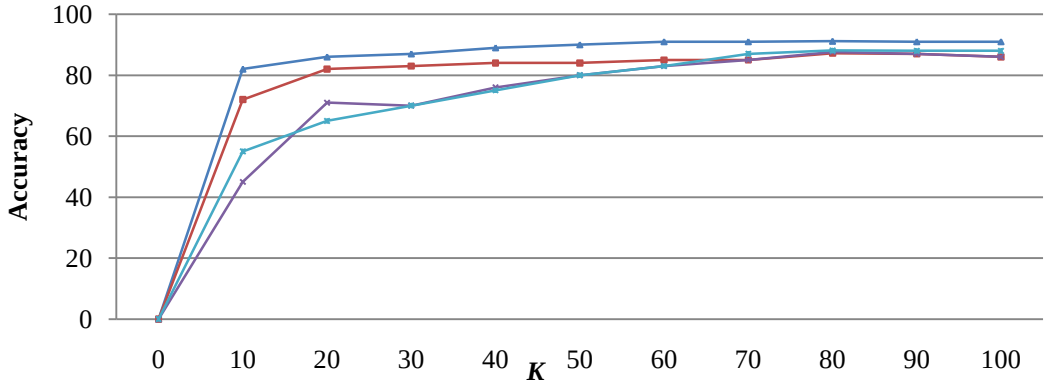


FIGURE 4.3: THE ACCURACY OF FOUR SETTINGS OF FNN_ MSFRB USING DIFFERENT VALUE OF K

Furthermore, the refinement step n in FNN_MSFBR settings is set from 2 to 30. Figure 4.4 demonstrates all four settings of FNN_MSFBR to achieve convergence within 22 steps and then their accuracy remains consistent. We set $n = 22$ because it can be implied that 22 iterations is enough to obtain the best accuracy in the MSFBR algorithm.

Figure 4.5 shows the average accuracy of four settings of FNN_MSFBR on three data sets by applying different values of α. It shows that 0.65 is the best value for α, which is selected in this paper.

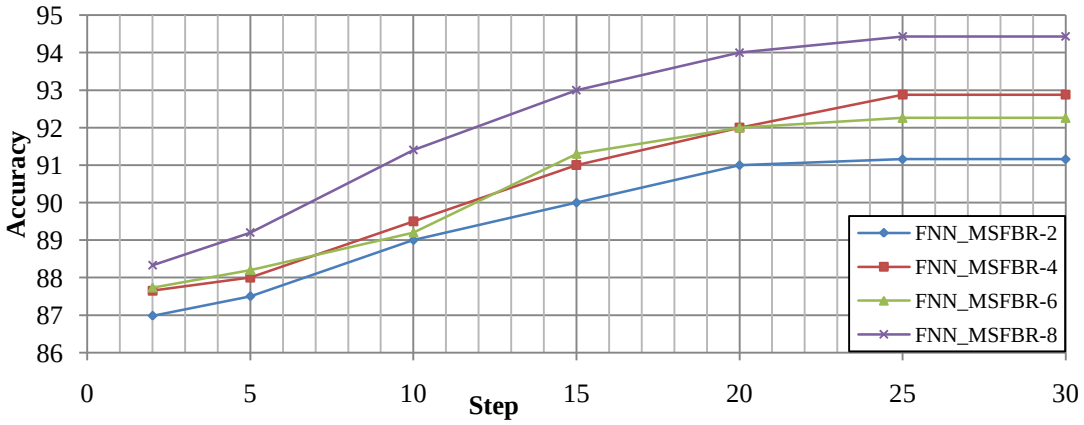


FIGURE 4.4: THE ACCURACY OF FOUR SETTINGS OF FNN_ MSFRB USING DIFFERENT STEPS

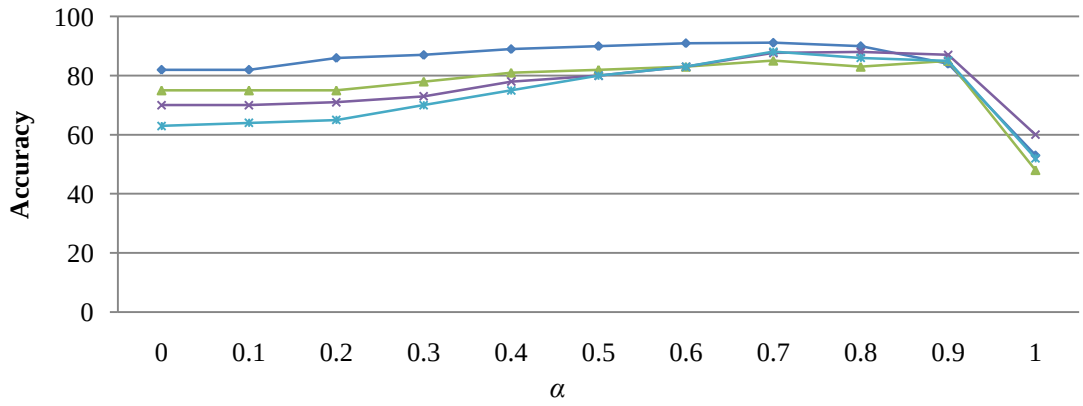


FIGURE 4.5: THE ACCURACY OF FNN_MSFBR USING DIFFERENT VALUE OF $\boldsymbol{\alpha}$

4.5 SUMMARY

In this chapter a Multi-Step Fuzzy Bridged Refinement-based (MSFBR) algorithm is proposed to solve the domain adaptation problem and applied to bank failure prediction. This is the first to utilize fuzzy set techniques to handle vague values of instance features in domain adaptation. Moreover, instead of modifying the baseline model or decision boundary, this study introduces fuzzy similarity/dissimilarity–based learning method as a local learning for domain adaptation. It explores similar/dissimilar fuzzy instances in the bridged domains and then, using the explored instances, refines the pseudo labels in the test data set that were initially predicted by the prediction model. Fuzzy neural network (Behbood et al. 2010) is considered as a

predictor to determine the initial labels for target instances. Sixteen experiments were performed using 20 years of bank failure financial data to evaluate the MSFBR algorithm, and to compare it with existing domain adaptation models. The results demonstrate that the MSFBR significantly outperforms other models in terms of long-term predictive accuracy. The outputs conclude that the MSFBR algorithm has interesting potential for implementation in financial applications for long-term bank failure prediction.

It must be emphasized that even though domain adaptation and concept drift are studied as the same technique in many database and data mining researches, they are very different. One of the major differences is that the entire training and test data are available to learn in domain adaptation while there is only a small number of test data for learning in concept drift (Yang 2009). Since we assume that all training and test data is available, the proposed algorithm is categorized as domain adaptation.

CHAPTER 5

FUZZY FEATURE ALIGNMENT-BASED CROSS-DOMAIN ADAPTATION

5.1 INTRODUCTION

Although machine learning technologies have gained a remarkable level of attention in researches in different computational fields, including prediction, most of them work under the common assumption that the training data (source domain) and the test data (target domain) have identical feature spaces with underlying distribution. As a result, once the feature space or the feature distribution of the test data changes, the prediction models cannot be used and must be rebuilt and retrained from scratch using newly-collected training data, which is very expensive if not practically impossible (Pan & Yang 2010). Similarly, since learning-based models need adequate labeled data for training, it is nearly impossible to establish a learning-based model for a domain (target domain) which has very few labeled data available for supervised learning. If we can transfer and exploit the knowledge from an existing similar but not identical domain (source domain) with plenty of labeled data, however, we can pave the way for construction of the learning-based model for the target domain. In real world scenarios, particularly in the finance industry, there are many situations in which very few labeled data are available, and collecting new labeled training data

and forming a particular model are practically impossible. For instance there are plenty of labeled data available to construct a prediction model to specify bank status in the state of California (source domain), whereas there are very few samples available for the banking system in the state of Texas (target domain). Since they might not have identical feature spaces, it is not possible to use the same model for both domains. However, they are similar and have common features, which may assist in the employment of the prediction model in the target domain. To transfer the knowledge between these two domains, we can explore the similarities, construct the cross-domain relationship between two domains with different but related feature spaces, and bridge the gap between two domains through this relationship.

Transfer learning methods have emerged in the computer science literature as a means of transferring knowledge from a source domain to a target domain. Transductive transfer learning is one category of transfer learning, in which the learning tasks are the same in both domains, while the source and target domains are different. It can be divided into two cases: (1) Domain adaptation assumes that the feature spaces of both domains are similar but that the marginal probability distribution of the data is different; and (2) Cross-domain adaptation assumes that the feature spaces are different in both domains but that they have some features in common. This chapter aims to solve the problem of cross-domain adaptation.

Most existing studies in transductive transfer learning focus on the domain adaptation problem and few researches have investigated the cross-domain adaptation problem. Even these few cross-domain adaptation studies have only focused on the cross-domain text classification problem using probabilistic models. Transfer learning, particularly cross-domain adaptation, which is a new machine learning and data mining framework, can be implemented in many novel applications, but most studies have been conducted in text classification and reinforcement learning and there is a lack of published novel applications of transfer learning in other areas (Yang 2009). This chapter will explore the cross-domain adaptation problem in bank failure prediction. Despite the recent surge of research in cross-domain adaptation, certain

issues have still not been taken into account and remain as challenges, such as handling the vagueness in feature values using soft computing methods, selecting significant features instead of instances in the target domain, and specifying an explicit relation among domains to construct a more general and independent model. We therefore develop a novel fuzzy cross-domain adaptation approach in this chapter to overcome these issues.

Three feature spaces are first defined: domain-independent; source domain-specific and target-domain specific. Based on these three feature spaces, the proposed approach is conducted in five main phases:

(1) A FNN is trained using labeled source instances based on source domain features and the initial labels of target instances are predicted by this model based on a domain-independent feature space;

(2) Target instances' labels are refined using MSFBR algorithm (Behbood et al. 2011) in Chapter 4;

(3) A Fuzzy Genetic Feature Weighting algorithm (Ramze Rezaee et al. 1999; Rhee & Lee 1999) is applied to weight the target-domain specific features using refined labels;

(4) A fuzzy spectral feature alignment algorithm is applied to cluster the features and then weight each feature in the target domain based on their correlation; and

(5) The significant target-domain specific features are selected according to the gained weights and the fuzzy prediction model is retrained using refined labels.

The main contributions of the proposed fuzzy cross-domain adaptation approach are: (1) It is capable of handling the uncertainty issue in data sets, while most existing transfer learning models work well with numerical crisp values; (2) The approach establishes an explicit relation between domains by specifying the significant features instead of instances in the target domain. The features are selected based on two weights achieved from domain-independent features which are similar in both domains but have different marginal distributions, and on domain-specific features which are different but have significant correlation; (3) The approach focuses on

currently given training and test data rather than the baseline model and decision boundary, and is thus more general and independent of the prediction model; and (4) The proposed approach is general, such that it can be applied for bank failure prediction and is not specified to particular applications such as natural language processing and text classification – a point which is worth emphasizing.

This chapter is organized as follows: Section 5.2 introduces the setting of problem we aim to solve. Section 5.3 outlines the fuzzy cross-domain adaptation approach and explains each of its phases in detail. Section 5.4 presents the evaluation and analysis of the experimental results for bank failure prediction. Finally, the chapter summary is discussed in Section 5.5.

5.2 PROBLEM SETTING AND DEFINITIONS

In this section the cross-domain adaptation problem, which this study intends to solve, is explained in detail and the notations to be used throughout the chapter are introduced.

Definition 5.1 (Source Domain)

$$D_s = \left\{F = (F_1, \dots F_n), \mu_F = \left(\mu_{F_1}, \dots, \mu_{F_n}\right) | \mu_{F_i} = \left(\mu_{f_{i1}}, \dots, \mu_{f_{ik_i}}\right)\right\},$$

where F is the feature space vector, μ_F is the membership function vector of features and $\mu_{f_{ij}}$ is the membership function of j-th linguistic term of i-th feature.

Definition 5.2 (Target Domain)

$$D_t = \left\{E = (E_1, \dots, E_m), \mu_E = \left(\mu_{E_1}, \dots, \mu_{E_m}\right) | \mu_{E_i} = \left(\mu_{e_{i1}}, \dots, \mu_{e_{ik_i}}\right)\right\},$$

where E is the feature space vector, μ_E is the membership function vector of features and $\mu_{e_{ij}}$ is the membership function of j-th linguistic term of i-th feature.

Definition 5.3 (Source Task)

$$T_s = \{L = (l_1, \dots, l_z), \mu_L = \left(\mu_{l_1}, \dots, \mu_{l_z}\right) | \mu_L(X) = FNN(X), X \in D_s\},$$

where $L$is the label space , μ_L is the membership function vector of labels and *FNN* is a predictive function which is not observed and to be learned by pairs

$$\left(X(t), \mu_L\left(X(t)\right)\right), t = 1, \dots, N_s,$$

where N_s is the population of instances in the source domain.

Definition 5.4 (Target Task)

$T_t = \{L = (l_1, \dots, l_z), \mu_L = (\mu_{l_1}, \dots, \mu_{l_z} | \mu_L(X) = FNN(X), X \in D_t\}$,

where L is the label space, μ_L is the membership function vector of labels and FNN is a predictive function which is not observed and to be learned by pairs

$\left(X(t), \mu_L(X(t))\right), t = 1, \dots, N_t$,

where N_t is the population of instances in the target domain.

Unified Feature Space: $UFS = \{G_i | G_i \in F \cup E\}$

Domain-Independent Feature Space: $DIFS = \{H_i | H_i \in F \cap E\}$

Domain-Specific Feature Space: $DSFS = \{\Psi_i | \Psi_i \in UFS - DIFS\}$

Source Domain-Specific Feature Space: $DSFS_s = \{\Phi_i | \Phi_i \in F - DIFS\}$

Target Domain-Specific Feature Space: $DSFS_t = \{\Omega_i | \Omega_i \in E - DIFS\}$

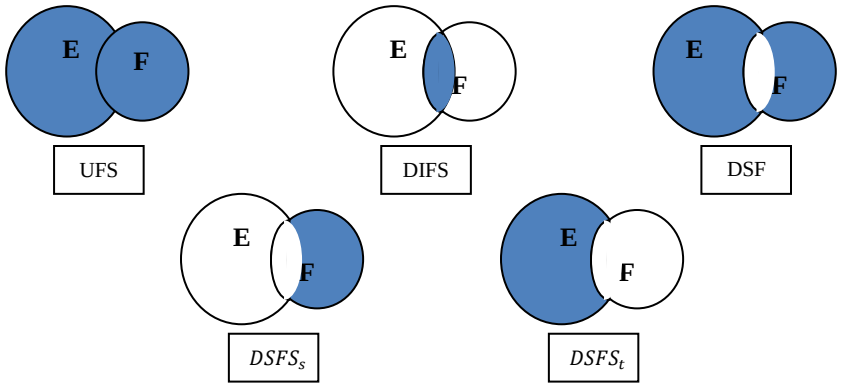


FIGURE 5.1: DIAGRAMS OF DOMAINS

Definition 5.5 (Cross-Domain Adaption) This is a category of transductive transfer learning in which $T_s = T_t$ and $D_s \neq D_t$, particularly $F \neq E$ but $DIFS \neq \emptyset$ and $\exists h_i \in DIFS$, $\mu_{h_i \in F} \neq \mu_{h_i \in E}$. In this situation, no labeled data are available in D_t while a quantity of labeled data are available in the source domain. The proposed approach aims to predict the instances' labels in D_t using knowledge from D_s.

Example 5.1 We provide an example to demonstrate the problem setting and use it to simplify the understanding of the proposed approach in the rest of the paper. The example consists of source and target domains with 100 instances in each domain. The source domain has three fuzzy features $\{F_1, F_2, F_3\}$ such that $\{F_1, F_2\} \in DIFS$ and $\{F_3\} \in DSFS_s$. The target domain includes five fuzzy features $\{E_1, E_2, E_3, E_4, E_5\}$ such that $\{E_2, E_4\} \in DIFS$ and $\{E_1, E_3, E_5\} \in DSFS_t$. All fuzzy features in the source and target domains have three or five linguistic terms which are formulated based on the data of instances in each domain. According to the problem definition and the example provided, it can be concluded that:

Unified Feature Space:

$$UFS = \{G_i | G_i \in F \cup E\} = \{G_1, G_2, G_3, G_4, G_5, G_6\}$$
$$= \{F_1 \cong E_4, F_2 \cong E_2, F_3, E_1, E_3, E_5\}.$$

Domain-Independent Feature Space:

$$DIFS = \{H_i | H_i \in F \cap E\} = \{H_1, H_2\} = \{F_1 \cong E_4, F_2 \cong E_2\}.$$

Domain-Specific Feature Space:

$$DSFS = \{\Psi_i | \Psi_i \in UFS - DIFS\} = \{\Psi_1, \Psi_2, \Psi_3, \Psi_4\} = \{F_3, E_1, E_3, E_5\}.$$

Source Domain-Specific Feature Space:

$$DSFS_s = \{\Phi_i | \Phi_i \in F - DIFS\} = \{\Phi_1\} = \{F_3\}.$$

Target Domain-Specific Feature Space:

$$DSFS_t = \{\Omega_i | \Omega_i \in E - DIFS\} = \{\Omega_1, \Omega_2, \Omega_3\} = \{E_1, E_3, E_5\}.$$

The following equations and figures show the fuzzy features and their linguistic terms' membership functions.

$$D_s = \{F = (F_1, F_2, F_3), \mu_F = (\mu_{F_1}, \mu_{F_2}, \mu_{F_3})\}$$

$$\mu_{F_1} = (\mu_{f_{11}}, \mu_{f_{12}}, \mu_{f_{13}})$$

$$\mu_{f_{11}} = \begin{cases} 2x - 1, & \text{if} \quad 0.5 \le x \le 1 \\ -2x + 3, & \text{if} \quad 1 \le x \le 1.5 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{f_{12}} = \begin{cases} 2x-2, & \text{if} \quad 1 \le x \le 1.5 \\ -2x+4, & \text{if} \quad 1.5 \le x \le 2 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{f_{13}} = \begin{cases} 2x-3, & \text{if} \quad 1.5 \le x \le 2 \\ -2x+5, & \text{if} \quad 2 \le x \le 2.5 \\ 0, & \text{if} \quad O.W \end{cases}$$

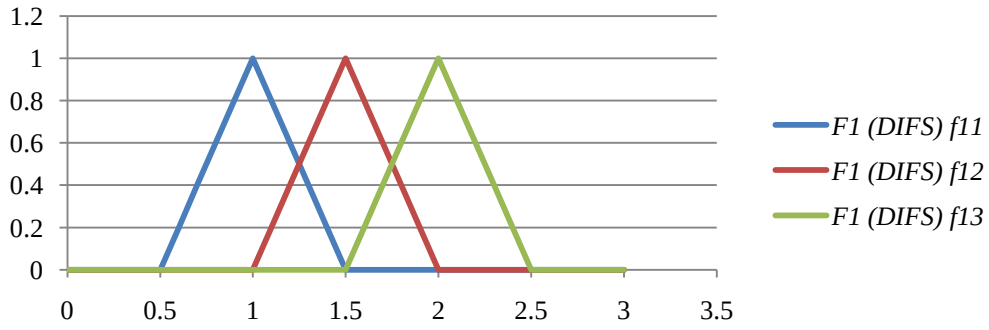


FIGURE 5.2: MEMBERSHIP FUNCTION OF FEATURE ONE OF SOURCE DOMAIN

$$\mu_{F_2} = (\mu_{f_{21}}, \mu_{f_{22}})$$

$$\mu_{f_{21}} = \begin{cases} x-10, & \text{if} \quad 1.5 \le x \le 2 \\ 1, & \text{if} \quad 2 \le x \le 2.5 \\ -x+13, & \text{if} \quad 12 \le x \le 13 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{f_{22}} = \begin{cases} 0.5x-6, & \text{if} \quad 12 \le x \le 14 \\ -0.5x+8, & \text{if} \quad 14 \le x \le 16 \\ 0, & \text{if} \quad O.W \end{cases}$$

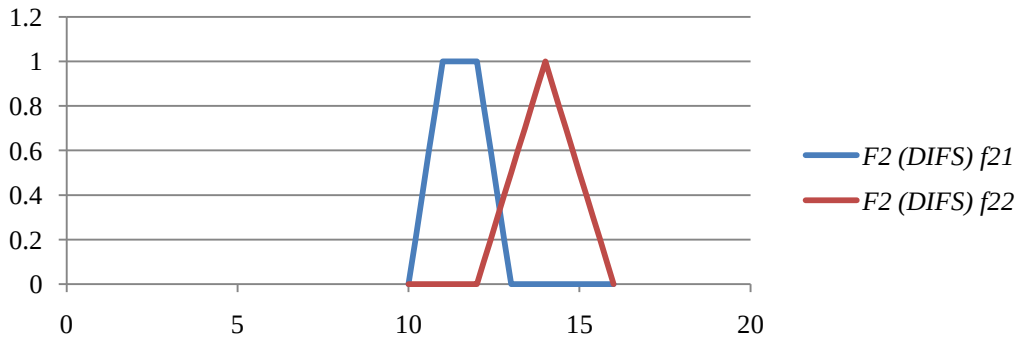


FIGURE 5.3: MEMBERSHIP FUNCTION OF FEATURE TWO OF SOURCE DOMAIN

$$\mu_{F_3} = (\mu_{f_{31}}, \mu_{f_{32}}, \mu_{f_{33}})$$

$$\mu_{f_{31}} = \begin{cases} 0.5x - 0.5, & \text{if} \quad 1 \le x \le 3 \\ 1, & \text{if} \quad 3 \le x \le 5 \\ -0.5x + 3.5, & \text{if} \quad 5 \le x \le 7 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{f_{32}} = \begin{cases} x - 6, & \text{if} \quad 6 \le x \le 7 \\ 1, & \text{if} \quad 7 \le x \le 8 \\ -x + 9, & \text{if} \quad 8 \le x \le 9 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{f_{33}} = \begin{cases} x - 9, & \text{if} \quad 9 \le x \le 10 \\ 1, & \text{if} \quad 10 \le x \le 12 \\ -x + 13, & \text{if} \quad 12 \le x \le 13 \\ 0, & \text{if} \quad O.W \end{cases}$$

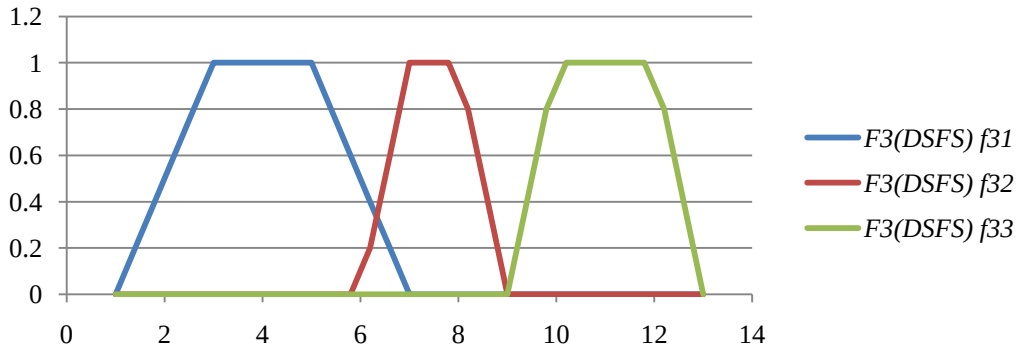


FIGURE 5.4: MEMBERSHIP FUNCTION OF FEATURE THREE OF SOURCE DOMAIN

$$D_t = \{E = (E_1, E_2, E_3, E_4, E_5), \mu_E = (\mu_{E_1}, \mu_{E_2}, \mu_{E_3}, \mu_{E_4}, \mu_{E_5})\}$$

$$\mu_{E_1} = (\mu_{e_{11}}, \mu_{e_{12}}, \mu_{e_{13}})$$

$$\mu_{e_{11}} = \begin{cases} \dfrac{x - 100}{50}, & \text{if} \quad 100 \le x \le 150 \\ \dfrac{-x + 200}{50}, & \text{if} \quad 150 \le x \le 200 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{e_{12}} = \begin{cases} \dfrac{x-150}{30}, & \text{if} \quad 150 \le x \le 180 \\ 1, & \text{if} \quad 180 \le x \le 320 \\ \dfrac{-x+350}{30}, & \text{if} \quad 1.5 \le x \le 2 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{e_{13}} = \begin{cases} (x-300)/50, & \text{if} \quad 1300 \le x \le 350 \\ (-x+400)/50, & \text{if} \quad 350 \le x \le 400 \\ 0, & \text{if} \quad O.W \end{cases}$$

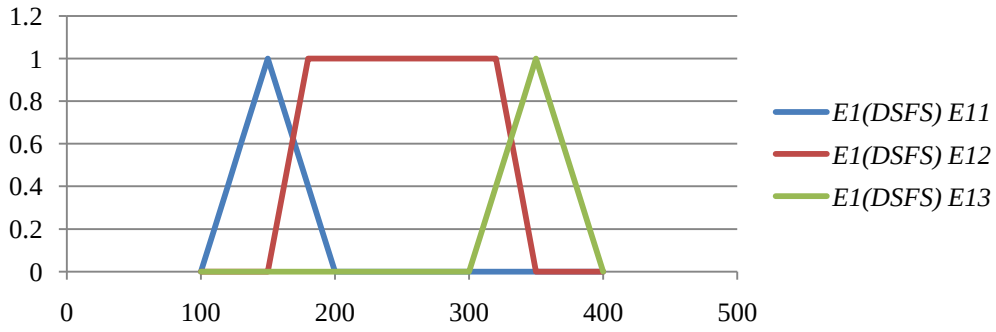


FIGURE 5.5: MEMBERSHIP FUNCTION OF FEATURE ONE OF TARGET DOMAIN

$$\mu_{E_2} = (\mu_{e_{21}}, \mu_{e_{22}})$$

$$\mu_{e_{21}} = \begin{cases} \dfrac{2(x-10)}{3}, & \text{if} \quad 10 \le x \le 11.5 \\ \dfrac{2(-x+13)}{3}, & \text{if} \quad 11.5 \le x \le 13 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{e_{22}} = \begin{cases} 0.5x-6, & \text{if} \quad 12 \le x \le 14 \\ -0.5x+8, & \text{if} \quad 14 \le x \le 16 \\ 0, & \text{if} \quad O.W \end{cases}$$

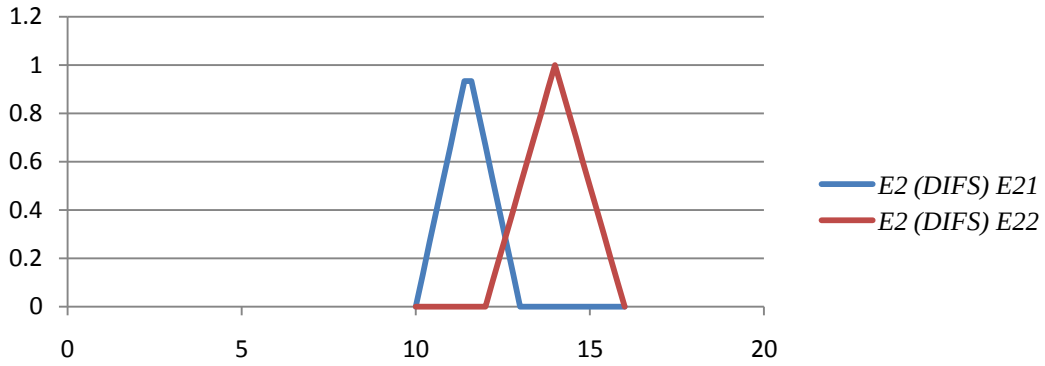


FIGURE 5.6: MEMBERSHIP FUNCTION OF FEATURE TWO OF TARGET DOMAIN

$$\mu_{E_3} = (\mu_{e_{31}}, \mu_{e_{32}}, \mu_{e_{33}})$$

$$\mu_{e_{31}} = \begin{cases} 10x, & \text{if} \quad 0 \le x \le 0.1 \\ 1, & \text{if} \quad 0.1 \le x \le 0.3 \\ -10x + 4, & \text{if} \quad 0.3 \le x \le 0.4 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{e_{32}} = \begin{cases} 10x - 3, & \text{if} \quad 0.3 \le x \le 0.4 \\ 1, & \text{if} \quad 0.4 \le x \le 0.6 \\ -10x + 7, & \text{if} \quad 0.6 \le x \le 0.7 \\ 0, & \text{if} \quad O.W \end{cases}$$

$$\mu_{e_{33}} = \begin{cases} 10x - 6, & \text{if} \quad 0.6 \le x \le 0.7 \\ 1, & \text{if} \quad 0.7 \le x \le 0.9 \\ -10x + 10, & \text{if} \quad 0.9 \le x \le 1 \\ 0, & \text{if} \quad O.W \end{cases}$$

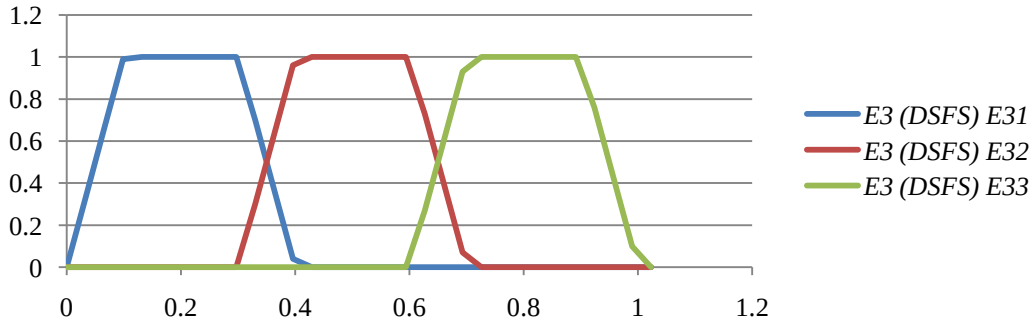


FIGURE 5.7 : MEMBERSHIP FUNCTION OF FEATURE THREE OF TARGET DOMAIN

$$\mu_{E_4} = (\mu_{e_{41}}, \mu_{e_{42}}, \mu_{e_{43}})$$

$$\mu_{e_{41}} = \begin{cases} 10(x - 0.5)/4, & \text{if } 0.5 \le x \le 0.9 \\ 1, & \text{if } 0.9 \le x \le 1.1 \\ 10(-x + 1.5)/4, & \text{if } 1.1 \le x \le 1.5 \\ 0, & \text{if } O.W \end{cases}$$

$$\mu_{e_{42}} = \begin{cases} 10(x - 1.1)/3, & \text{if } 1.1 \le x \le 1.4 \\ 1, & \text{if } 1.4 \le x \le 1.6 \\ 10(-x + 1.9)/3, & \text{if } 1.6 \le x \le 1.9 \\ 0, & \text{if } O.W \end{cases}$$

$$\mu_{e_{43}} = \begin{cases} 2x - 3, & \text{if } 1.5 \le x \le 2 \\ -2x + 5, & \text{if } 2 \le x \le 2.5 \\ 0, & \text{if } O.W \end{cases}$$

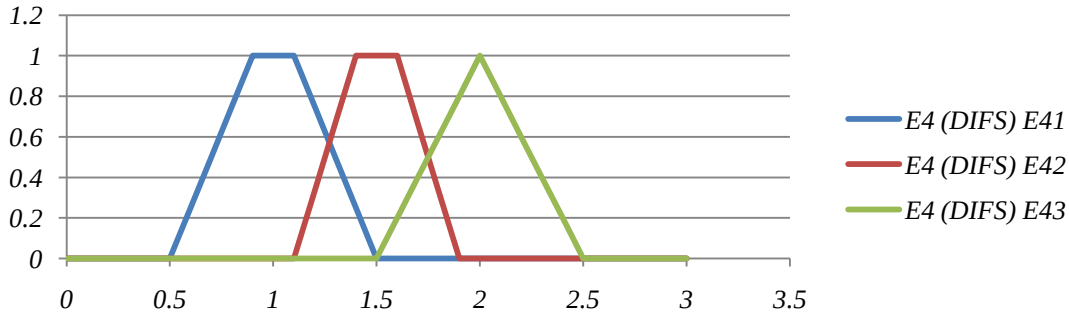


FIGURE 5.8: MEMBERSHIP FUNCTION OF FEATURE FOUR OF TARGET DOMAIN

$$\mu_{E_5} = (\mu_{e_{51}}, \mu_{e_{52}}, \mu_{e_{53}}, \mu_{e_{54}}, \mu_{e_{55}})$$

$$\mu_{e_{51}} = \begin{cases} x + 3, & \text{if } -3 \le x \le -2 \\ -x - 1, & \text{if } -2 \le x \le -1 \\ 0, & \text{if } O.W \end{cases}$$

$$\mu_{e_{52}} = \begin{cases} x + 2, & \text{if } -2 \le x \le -1 \\ -x, & \text{if } -1 \le x \le 0 \\ 0, & \text{if } O.W \end{cases}$$

$$\mu_{e_{53}} = \begin{cases} x + 1, & \text{if } -1 \le x \le 0 \\ -x + 1, & \text{if } 0 \le x \le 10 \\ 0, & \text{if } O.W \end{cases}$$

$$\mu_{e_{54}} = \begin{cases} x, & \text{if } 0 \le x \le 1 \\ -x + 2, & \text{if } 1 \le x \le 2 \\ 0, & \text{if } O.W \end{cases}$$

$$\mu_{e_{55}} = \begin{cases} x - 1, & \text{if } 0 \le x \le 1 \\ -x + 3, & \text{if } 1 \le x \le 2 \\ 0, & \text{if } O.W \end{cases}$$

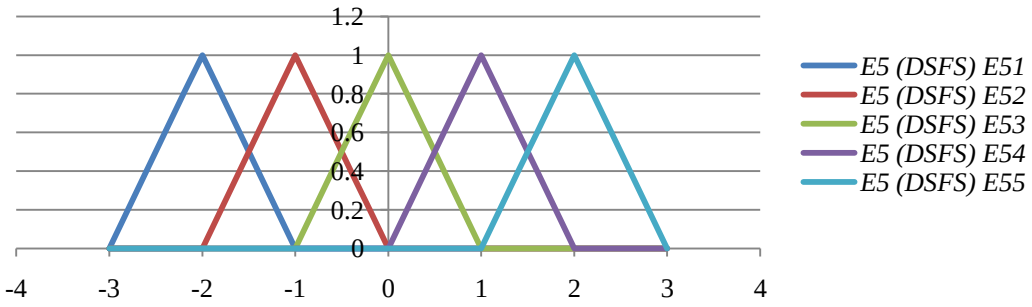


FIGURE 5.9: MEMBERSHIP FUNCTION OF FEATURE FIVE OF TARGET DOMAIN

5.3 THE FUZZY CROSS-DOMAIN ADAPTATION APPROACH

All five phases of the proposed approach, including the definitions and methods applied in each phase, are described in this section. Some phases are explained in greater detail by applying the example introduced in Section 5.2; in particular, the Fuzzy Spectral Feature Alignment (FSFA) algorithm, which is the main contribution of this chapter, is presented in Phase Three (Section 5.3.3). Figure 5.12 depicts the outline of the proposed approach.

5.3.1 PHASE ONE

Fuzzy Neural Network (FNN) (Behbood et al. 2010) is chosen as a prediction model in this phase. FNN is trained using labelled instances data in the source domain. The number of inputs of the FNN is equal to the number of features of the source domain and it has one output to assign a binary label to the instance. For instance, the FNN of Example 5.1 has three inputs corresponding to three features in the source domain and one output corresponding to a binary class label. The structure of the FNN is shown in Figure 5.10.

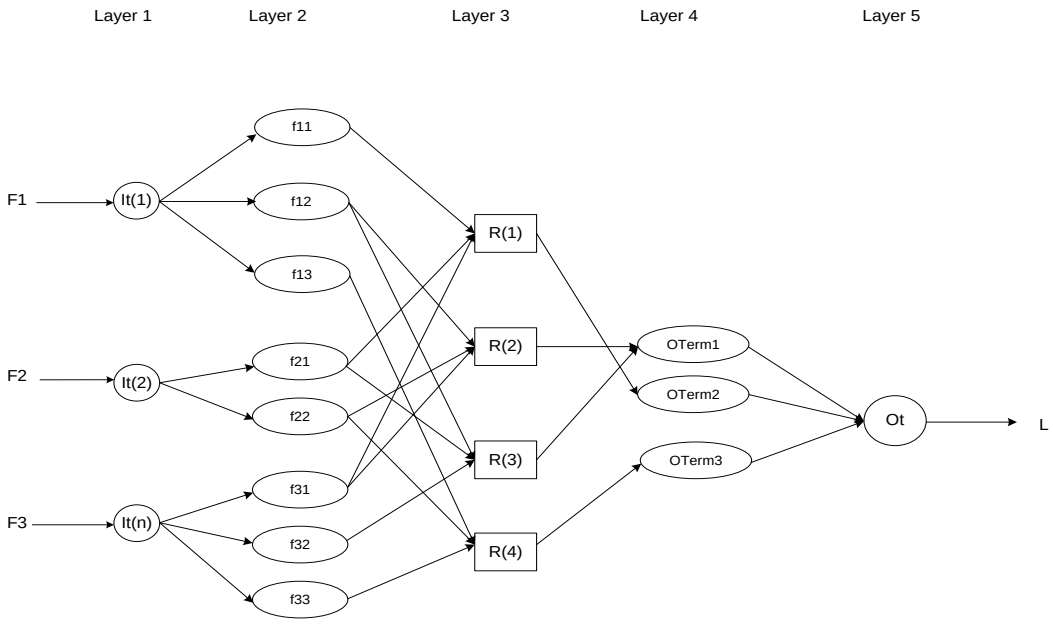


FIGURE 5.10: FNN STRUCTURE OF EXAMPLE 5.1 (PHASE 1)

5.3.2 PHASE TWO

The FNN trained in the previous phase is applied to predict the labels of the target instances using the data which is projected on Domain-Independent Feature Space (DIFS). This means that the number of inputs of FNN is the number of features in DIFS and the value of the target instances in these features is applied for prediction. The structure of the trained FNN of Example 5.1 for this phase is shown in Figure 5.11. Although the features in DIFS are similar in both domains, they have different distribution, so the MSFBR algorithm (Behbood et al. 2011) (proposed in Chapter 4) is applied to refine the target predicted labels and achieve better predictive accuracy. The MSFBR algorithm is a domain adaptation algorithm which is applied when source and target domains have similar features but their distributions are different.

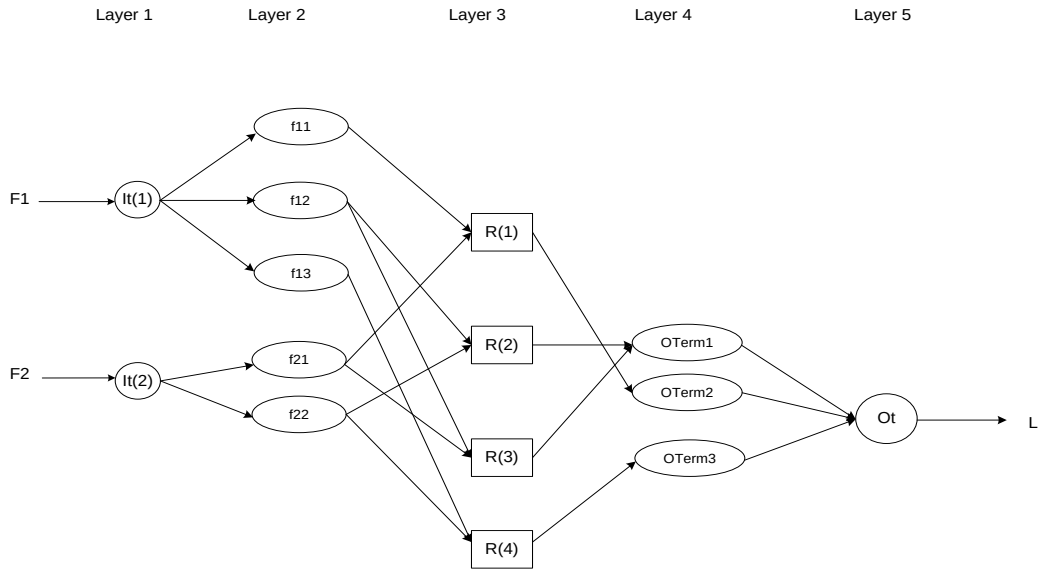


FIGURE 5.11: FNN STRUCTURE OF EXAMPLE 5.2 (PHASE 2)

5.3.3 PHASE THREE

In this phase, a new algorithm, called Fuzzy Spectral Feature Alignment (FSFA), to find the primary weight (ω_1) for fuzzy linguistic terms of DSFS is developed and applied. The primary weight (ω_1) demonstrates the significance of fuzzy linguistic terms of DSFS based on their relation to the fuzzy linguistic terms of DIFS. The primary weight (ω_1) is combined with the secondary weight (ω_2), which will be gained in Phase Four, to compute the final weight of each linguistic term.

5.3.3.1 FUZZY SPECTRAL FEATURE ALIGNMENT

One of the first clustering studies used the K-means algorithm, introduced by (Macqueen 1967). Its applicability is constrained since the clusters should be convex-shaped and well-separated which is usually violated in real world applications. Fuzzy C-mean was proposed by (Bezdek & Ehrlich 1984) as an alternative algorithm to avoid these limitations; however, its performance drastically decreased when applied

to non-convex-shaped clusters. To solve this problem, approaches based on spectral graph theory have recently been introduced for crisp and fuzzy clusters (Cominetti et al. 2010). These algorithms do not require a predefined number of clusters to run. The Fuzzy Spectral Graph Clustering algorithm (DifFUZZY) (Cominetti et al. 2010) merges the fuzzy clustering and spectral clustering to achieve their strengths and avoid their weaknesses. We apply DifFUZZY to propose a Fuzzy Spectral Feature Alignment algorithm (FSFA) to align the features in Domain-Specific feature spaces and then find the significant features in $DSFS_t$.

5.3.3.2 Fuzzy Spectral Feature Graph Structure

Based on the problem setting and given $DSFS_s$ and $DSFS_t$, a bipartite weighted graph (G) can be constructed: $G = (V_{DSFS} \cup V_{DIFS}, W)$ where each vertex in the first part corresponds to a particular fuzzy linguistic term of a feature in $DSFS$ and each vertex in the second part corresponds to a particular fuzzy linguistic term of a feature in $DIFS$: $\forall v_k \in \{V_{DSFS}\}: v_k \sim \mu_{\Psi_{ij}}(\cdot)$ and $\forall v_k \in \{V_{DIFS}\}: v_k \sim \mu_{\mathrm{h}_{ij}}(\cdot)$. Each edge in W connects two vertexes in V_{DSFS} and V_{DIFS} respectively and there is no connection between the vertexes in each part. Each edge $w_{ij} \in W$ has a weight which is computed based on the Fuzzy Correlation between two fuzzy linguistic terms in V_{DSFS} and V_{DIFS} which are connected by w_{ij}. Given $|V_{DSFS}|$ and $|V_{DIFS}|$are the number of linguistic terms in $DSFS$ and $DIFS$ respectively, the weight matrix of the proposed graph $W \in \mathbb{R}^{(|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|)}$ is formed as $W = \begin{bmatrix} 0 & FCC \\ FCC^T & 0 \end{bmatrix}$ where the first $|V_{DSFS}|$ rows and columns correspond to the fuzzy linguistic terms in $DSFS$, and the last $|V_{DIFS}|$ rows and columns correspond to the fuzzy linguistic terms in $DIFS$. FCC explained in the next section is the fuzzy correlation between the linguistic terms. The larger their correlation, the larger is the weight assigned to the edge. The proposed graph can be used to model the latent relationship between domain-independent features and domain-specific features, and to align the domain-specific features effectively by adapting spectral clustering on them.

5.3.3.3 FUZZY CORRELATION COEFFICIENT

To find the fuzzy correlation coefficient (FCC) between linguistic terms, the method introduced by Chiang (1999) is applied to assign weights to the edges of the graph. The method derives the following formula for FCC by adapting the concepts from conventional statistics. The value of FCC lies in the interval $[-1, 1]$ and has a similar meaning to the correlation coefficient in the conventional statistics. It represents the degree and type of relationship between fuzzy sets. The sign of the FCC demonstrates whether two sets are positively or negatively related.

$$FCC_{A,B} = \frac{\sum_{i=1}^{n}(\mu_A(x_i)-\bar{\mu}_A)(\mu_B(x_i)-\bar{\mu}_B)}{(n-1)(S_A S_B)}, \tag{5.1}$$

where $\bar{\mu}_A$ and $\bar{\mu}_B$ denote the average membership value of fuzzy sets A and B respectively. S_A and S_B are the standard deviation of fuzzy sets A and B respectively.

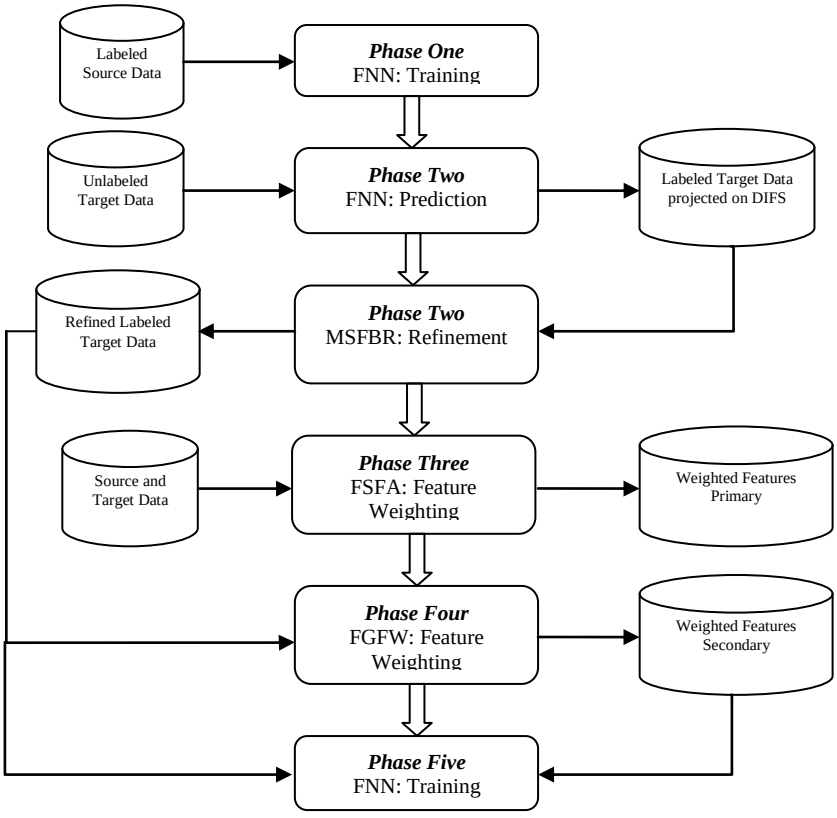


FIGURE 5.12: THE OUTLINE OF THE PROPOSED APPROACH

Table 5.1 presents the matrix of FCC for Example 5.1. The value of $FCCs$ between linguistic terms of $DSFS$ and $DIFS$ are assigned to edges of matrix as their weights and actually the FCC matrix forms the weight matrix (W) of the FSFA graph. Since $DSFS$ and $DIFS$ include 9 and 5 linguistic terms respectively in Example 5.1, the weight matrix of SFSA graph of Example 5.1 is formed as

$$W = \begin{bmatrix} {}^{|V_{DSFS}|=9} & \vdots & {}^{|V_{DIFS}|=5} \\ 0 & \vdots & |FCC| \\ \dots\dots\dots & \vdots & \dots\dots\dots \\ |FCC^T| & \vdots & 0 \end{bmatrix}_{14\times 14}$$

where $|FCC|$ is the absolute value of FCC.

TABLE 5.1: THE FCC FOR LINGUISTIC TERMS OF EXAMPLE 5.1

FCC	h11	h12	h21	h22	h23
Ψ11	0.706227	0.984089	0.666491	1	0.758657
Ψ12	0.204419	0.874219	0.433097	0.337182	0.854099
Ψ13	0.337101	0.454668	0.320385	0.477327	0.320385
Ψ21	0.706227	0.537321	0.310945	0.694990	0.537321
Ψ22	0.481451	0.490650	0.474595	0.538147	0.49065
Ψ23	0.440291	0.295527	0.310945	0.419390	0.295527
Ψ31	0.902998	0.661248	0.411883	0.903341	0.661248
Ψ32	0.179473	0.830254	0.206009	0.902405	0.400841
Ψ33	0.554483	0.517558	0.391590	0.528161	0.517558
Ψ41	0.706227	0.537321	0.310945	0.694990	0.537321
Ψ42	0.993916	0.404066	1	0.694990	0.404066
Ψ43	0.221907	0.976986	0.002488	1	0.295527
Ψ44	0.440291	0.861864	1	0.41939	1
Ψ45	0.440291	0.295527	0.310945	0.41939	0.295527

5.3.3.4 FUZZY SPECTRAL FEATURE ALIGNMENT ALGORITHM

In this section, the Fuzzy Spectral Feature Alignment (FSFA), which is based on graph spectral theory, is explained. Graph spectral theory assumes that if two nodes in a graph are connected to many common nodes, they should be similar. In the proposed algorithm, we assume that if two fuzzy linguistic terms of $DSFS$ are highly

connected to many common fuzzy linguistic terms of $DIFS$, they are most probably related to each other and will be aligned with a same cluster with a specific membership value. The proposed algorithm returns a set of membership values for each fuzzy linguistic term in each cluster. Based on the membership value, we can assign a weight to the most meaningful and significant features of $DSFS$ to reduce the gap between two domains and improve prediction accuracy. Before applying the algorithm, we need to find a number of initial parameters. First, we should identify the number of clusters N_c and their cores. Since we aim to cluster features of $DSFS$ based on their relation to features of $DIFS$, the number of clusters is equal to the number of linguistic terms of $DIFS$ ($N_c = |V_{DIFS}|$) and each linguistic term of $DIFS$ is considered as a core for each cluster.

< *Fuzzy Spectral Feature Alignment Algorithm*>

Input: *Source domain:* D_s

Target domain: D_t

Target Fuzzy feature space: $\tilde{F}_t$

Source Fuzzy feature space: $\tilde{F}_s$

Predictive fuzzy labels: $\tilde{Y}$

Domain-Independent Feature Space: $DIFS$

Domain-Specific Feature Space: $DSFS$

Source Domain-Specific Feature Space: $DSFS_s$

Target Domain-Specific Feature Space: $DSFS_t$

Prediction model $f(\cdot)$

Output: $\omega_1(\Psi_{ij}) = U_c(\Psi_{ij})$ *is the membership value of each fuzzy linguistic term of DSFS in each* cluster.

[Begin]

Step 1: Build Fuzzy Correlation Coefficient matrix

$$FCC_{|V_{DSFS}|\times|V_{DIFS}|} = \left[FCC_{\Psi_{ij},h_{ij}} \middle| FCC_{\Psi_{ij},h_{ij}} = \frac{\sum_{k=1}^{n}\left(\mu_{\Psi_{ij}}(x_k)-\overline{\mu}_{\Psi_{ij}}\right)\left(\mu_{h_{ij}}(x_k)-\overline{\mu}_{h_{ij}}\right)}{(n-1)\left(S_{\Psi_{ij}}S_{h_{ij}}\right)} \right]. \quad (5.2)$$

Step 2: Build FSFA Weight matrix (W)

$$W_{(|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|)} = \begin{bmatrix} \overbrace{0}^{|V_{DSFS}|} & \overbrace{FCC}^{|V_{DIFS}|} \\ FCC^T & 0 \end{bmatrix}_{(|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|)} \quad (5.3)$$

Step 3: Build Diagonal matrix (D)

$$D_{(|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|)} = \left[D_{i,j} = \begin{cases} \sum_{s=1}^{|V_{DSFS}|+|V_{DIFS}|} w_{p,s} & p = 1, \dots, |V_{DSFS}| + |V_{DIFS}|, \ i = j \\ 0 & , \ i \neq j \end{cases} \right]. \quad (5.4)$$

Step 4: Establish Transition matrix (P)

$$P = I + [W - D] \frac{\gamma_1}{\max_{i=1,\dots,|V_{DSFS}|+|V_{DIFS}|} D_{i,i}}, \quad (5.5)$$

where $I \in \mathbb{R}^{|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|}$ is the identity matrix and $\gamma_1 \in (0,1)$, which is an internal parameter, ensures that all entries of transition matrix are nonnegative. Its default value is 0.1.

Step 5: Establish Alternative matrixes

Let Ψ_o be a fuzzy linguistic term of $DSFS_t$. To assign its membership value ($U_c(\Psi_o)$) to cluster c_k, $k = 1, \dots, N_c$, an alternative weight matrix $\overline{W}$ is formed using original weight matrix W with the row and column of Ψ_o replaced by row and column of h_o which is the core of cluster c_k. Using $\overline{W}$, matrixes $\bar{D}$ and $\bar{P}$ are computed by (5.4) and (5.5) respectively.

Step 6: Calculate Diffusion Distance (DD)

$$\alpha = \left\lfloor \frac{\gamma_2}{|log\beta|} \right\rfloor, \quad (5.6)$$

where β is the second largest eigenvalue of P, $\lfloor \cdot \rfloor$ denotes the integer part and $\gamma_2 \in (0, \infty)$ is an internal parameter and its default value is 1.

$$DD(\Psi_o, c_k) = \begin{cases} \left|\left|P^\alpha e - \bar{P}^\alpha e\right|\right| \text{where} \quad e(j) = 1, & \text{if} \quad j = index(\Psi_o) \\ \left|\left|P^\alpha e - \bar{P}^\alpha e\right|\right| \text{where} \quad e(j) = 0, & \text{if} \quad O.W \end{cases}, \quad (5.7)$$

where $\left|\left|\cdot\right|\right|$ is the Euclidean norm and $k = 1, \dots, N_c$.

Step 7: Calculate Membership value

$$U_c(\Psi_o) = \{u_{c_k}(\Psi_o) | u_{c_k}(\Psi_o) = \frac{DD(\Psi_o, c_k)}{\sum_{l=1}^{N_c} DD(\Psi_o, c_l)}, k = 1, \dots, N_c\} \cdot \quad (5.8)$$

Repeat Steps 5 to 7 for each $\Psi_{ij} \in DSFS$.

[End]

5.3.4 PHASE FOUR

The algorithm to find the secondary weight (ω_2) is developed and described in this section. This weight, which represents the significance of the linguistic terms in prediction, will be combined with the primary weight (ω_1) in the next phase to find the final weight for each fuzzy linguistic term of $DSFS$.

5.3.4.1 FEATURE SELECTION

Feature selection is one of the most important steps of classification and prediction. Since some features are highly correlated and/or irrelevant to the objective task and also there might be many features in the feature space which make the computation complex and expensive, weighting and selecting the significant features are desirable. As a result of feature selection, the training procedure takes less time and the prediction model will obtain a higher generalization capability as a result of fewer features. In our case, detecting significant features in the target domain by using the labels predicted by similar features in the source domain helps us to transfer the knowledge from the source domain and find the important features based on the similarity between domains.

In a typical fuzzy classification or prediction model, each feature is represented by a number of fuzzy linguistic terms like LARGE, MEDIUM and SMALL, and the model is explicitly explained by a number of fuzzy if-then rules, such as:

If X is SMALL then Y is MEDIUM where X and Y are features.

Hence, weighting the fuzzy linguistic terms of each feature based on their importance in prediction forms the optimum prediction model. In the method proposed by Rezaee et al. (1999), which is applied in the proposed approach, an optimal subset of fuzzy linguistic terms is selected by using conventional search techniques. Instead of a conventional research technique, we use a faster and more efficient weighting method based on a fuzzy genetic approach proposed by Rhee and Lee (1999).

5.3.4.2 FUZZY GENETIC FEATURE WEIGHTING ALGORITHM

The proposed Fuzzy Genetic Feature Weighting (FGFW) algorithm applies the instances in the target domain with the labels which were predicted by FNN-MSFBR in Phase Two to assign a secondary weight (ω_2) to each fuzzy linguistic termof . It is described in detail as follows:

<Fuzzy Genetic Feature Weighting Algorithm>

Input: *Refined labels of target instances computed in Phase 2 :*
Population Number: PN
Crossover Probability:
Mutation Probability: MP
Bit Length: BL
Selection Threshold: ST
Error Threshold: ET

Output: *is the weight value of each fuzzy linguistic term in*

[Begin]

Step 1: Initialization

Initialize algorithm parameters: PN

Step 2: Project the original data to fuzzy space and generate random population

(5.9)

then a random population of PN chromosomes is generated. Each chromosome, which is representative of each fuzzy linguistic term, consists of BL gens.

For $s = 1$ to PN

Step 3: Calculate the secondary weight and compute the fitness function value

3.1: $J^s(\mathrm{e}_{ij}) =$ $\frac{hc/r}{BL}$ · (5.10)

3.2: Execute the FNN prediction model with lingustic terms {e

using $\{\left(X(t), \mu_L(X(t))\right) | X(t) \in D_t$,

3.3: Calculate fitness function:

$$E(s) = \big(1 - G(s)\big) = (1 - \sqrt{\frac{TP(s)}{TP(s)\times FN(s)} \times \frac{TN(s)}{FP(s)\times TN(s)}}) \cdot \tag{5.11}$$

Next s

Repeat

Step 4: Reproduction

4.1: Select two parent chromosomes according to their values of $E(i)$. (the lower the value of $E(i)$, the more probability there is that it will be selected)

4.2: Cross over the selected parents using *CP* to form a new offspring.

4.3: Mutate new offspring at each locus using *MP*.

4.4: Place new offspring in a new population.

Until $|\text{new population}| < NP$

If $\exists s,\ E(s) < ET$

Then return $\omega_2(\mathrm{e}_{ij}) = J^s(\mathrm{e}_{ij})$

Else go to Step 3

[End]

5.3.5 PHASE FIVE

In this Phase, the primary and secondary weights gained in previous phases are combined to achieve the final weight for each fuzzy linguistic term in the target domain. According to the final weight, most significant linguistic terms are selected to train the FNN and make the final prediction. The final weight is computed using Equation 5.12 where β is the experimental trade-off parameter to find the optimal combination of primary and secondary weights. The value of β is empirically achieved in each experiment to achieve maximum accuracy.

$$\omega = \beta\omega_1 + (1-\beta)\omega_2 \cdot \tag{5.12}$$

The final weight (ω) of each fuzzy linguistic term is compared with a predefined threshold (ε) and if it satisfies the threshold ($\omega > \varepsilon$), then the corresponded linguistic terms are selected to participate in training the prediction model. The selected

linguistic term is denoted as the Selected Target-Domain Feature Space (STDFS) and is presented as $\Lambda = \{\Lambda_i | \ \omega(\Lambda_i) > \varepsilon\}$**.** The FNN prediction model is trained using the labeled target instances data (gained in Phase Two) which are projected on:

$$\Lambda: \left\{(\mu_\Lambda(x), \mu_L(x)) | \mu_\Lambda(x) = \left\{\mu_{\Lambda_1}(x), \dots, \mu_{\Lambda_q}(x)\right\}, \omega(\Lambda_i) > \varepsilon \, , \mu_L(x) = FNN\_MSFBR(x), x \in D_t\right\}$$

5.4 Empirical Analysis for Bank Failure Prediction

In this section we validate the proposed fuzzy cross-domain adaptation approach using synthetic and real world financial data. The task in this experiment is binary classification for synthetic data and bank failure prediction for real world financial data in which the prediction label has two classes: Failed or Survived. We perform a number of experiments to examine the performance approach to find the significant features in the target domain through similar features in both domains and consequently predict the labels of instances in the target domain. The predictive accuracy of the proposed approach is examined using different baselines and is benchmarked with other similar existing methods. The results demonstrate a significant improvement which is proved by statistical tests.

5.4.1 Data Sets

The data sets used in the experiments are divided into two groups: synthetic data set and real data set. The first data set is created by the authors while the second is derived from a financial institution. In the following sections, these two data sets are explained in detail.

5.4.1.1 Synthetic Data Set

The data set consists of two tables of data: source domain table and target domain table. Each table includes 2000 instances. The number of features in the source domain table and the target domain table are 10 and 15 respectively. Three features, which belong to the $DIFS$, are similarly designed with the same number of linguistic

terms, but they are slightly different in their fuzzy membership functions of linguistic terms. All features in the source domain are designed to have a significant impact on prediction output, while 12 out of 15 features in the target domain are designed to have a meaningful relation to the instances' labels with different degree. The authors aim to find the significant features in the target domain using the features in the source domain. In each domain, 15% (300) of instances are labeled as negative instances, which we are interested in recognizing. This data set is summarized in Table 5.2. According to the design of the data set, the following information is established:

Number of features in UFS: $N_{UFS} = 22$

Number of features in $DIFS$: $N_{DIFS} = 3$

Number of features in $DSFS$: $N_{DSFS} = 19$

Number of features in $DSFS_s$: $N_{DSFS_s} = 7$

Number of features in $DSFS_t$: $N_{DSFS_t} = 12$

TABLE 5.2: SYNTHETIC DATA SET

Domain	N.O. Features	N.O. Instances	N.O. Negative	N.O. Positive
Source	10	2000	300 (15%)	1700(85%)
Target	15	2000	300(15%)	1700(85%)

5.4.1.2 REAL DATA SET

The data set and financial variables are extracted from Call Report Data downloaded from the website of the Federal Reserve Bank of Chicago[10] and the status of each bank is identified according to the Federal Financial Institutions Examination Council[11]. The data set includes the observation period of the survived banks of 21 years from Jun 1980 to Dec 2000 and is based on the history of each bank in FFIEC. The authors collected the history data of banks in 12 different States including: *TX; IL; MN; IA CA; KS; MO; GA* and *FL* for three different years: 1995, 1998 and 2000. Each State has a different number of banks (instances) which are categorized in two divisions: Failed and Survived. Fewer portions of whole data (on average 16%) in

[10] http://www.chicagofed.org

[11] http://www.ffiec.gov/nicpubweb/nicweb/NicHome.aspx

each State and year are failed banks, which the authors are interested in predicting, and thus the imbalanced data set problem arises. This problem will be solved in experiments. Although Tung et al. (2004) used nine financial features, according to their statistical significance and correlation, it is observed that the FNN with three features has fewer created rules, less computational load and greater prediction accuracy (Ng et al. 2008). The definitions of all features are described in Table 2.2. In each experiment, one State called the Target-State is considered as the target domain with nine features, and one or more other States called Source-States are selected as source domains with three features. The authors would like to weight the features in the Target-State based on features in the Source-States to predict the failed banks in three different time periods: 5-year, 2-year and same-year. The historical data for 1995, 1998 and 2000 are applied to train the FNN and then predict bank status in 2000. The data set is summarized in Table 5.3. According to the data set the following information can be derived:

Number of features in UFS: $N_{UFS} = 9$

Number of features in $DIFS$: $N_{DIFS} = 3$

Number of features in $DSFS$: $N_{DSFS} = 6$

Number of features in $DSFS_s$: $N_{DSFS_s} = 0$

Number of features in $DSFS_t$: $N_{DSFS_t} = 6$

5.4.2 RESEARCH DESIGN

To reduce the influence of the imbalanced data set problem, the SMOTE technique (Chawla et al. 2002) is applied to training data sets. The number of failed banks increases in relation to the number of survived banks to achieve a balanced data set, which improves the accuracy of prediction without losing important information. In each experiment, the training data set splits into two pools: (1) Negative instances (failed banks) denoted with label 1; (2) Positive instances (survived banks) denoted with label 0. The 5-fold cross validation method is applied for training. The predictors are trained using training data sets and are then evaluated by the testing data sets. The

predictive accuracy in each experiment, which is the mean value of cross-validation group accuracy, is calculated using GM.

TABLE 5.3: REAL WORLD FINANCIAL (BANK FAILURE) DATA SET

State	2000			1998			1995		
	Total	Survived Banks	Failed Banks	Total	Survived Banks	Failed Banks	Total	Survived Banks	Failed Banks
IL	794	665 (83.75%)	129	840	698 (83.09%)	142	971	796 (81.98%	175
TX	743	632 (85.06%)	111	835	716 (85.75%)	119	967	813 (84.07%)	154
MN	493	406 (82.35%)	87	515	419 (81.36%)	96	526	430 (81.75%)	96
IA	431	380 (88.17%)	51	443	365 (82.39%)	78	491	426 (86.76%)	65
CA	381	322 (84.51%)	59	432	377 (87.27%)	55	511	428 (83.76%)	83
NY	379	314 (82.85%)	65	423	354 (83.69%)	69	490	418 (85.31%)	72
KS	376	306 (81.38%)	70	393	336 (85.50%)	57	433	367 (84.76%)	66
MO	363	314 (86.50%)	49	384	329 (85.68%)	55	462	385 (83.33%)	77
GA	346	288 (83.24%)	58	364	301 (82.69%)	63	401	332 (82.79%)	69
WI	335	277 (82.69%)	58	368	306 (83.15%)	62	411	338 (82.23%)	73
FL	311	262 (84.24%)	49	306	257 (83.99%)	49	392	321 (81.89%)	71
OK	286	245 (85.66%)	41	309	262 (84.79%)	47	342	288 (84.21%)	54

5.4.2.1 EXPERIMENT DESIGN USING SYNTHETIC DATA SET

To evaluate the performance of the proposed approach and investigate the importance of each phase of the proposed approach, the following baselines are formed and compared to one another using the synthetic data:

(1) NoTra: The FNN trained by the source domain training data and applied to the target domain using features in DIFS. It is expected to be the worst baseline for evaluating the performance of other baselines.

(2) TraRef: The FNN when only Phase Two of the proposed approach is applied. It means that the FNN is trained by the source domain data and applied to the target domain using features in DIF. The labels are then refined by the MSFBR algorithm.

(3) TraRefWei1: The FNN when Phase Four of the proposed approach is ignored. It means that the FSFA algorithm is applied to weight the features, and the significant features in the target domain are selected based on ω_1 alone. The FNN is then trained using these features.

(4) TraRefWei2: The FNN when Phase Three of the proposed approach is ignored. It means that the FGFW algorithm is applied to weight the features and the significant features in the target domain are selected based on ω_2 alone. The FNN is then trained using these features.

(5) TraRefWei1&2: The proposed approach with all phases.

(6) TraRefWei1R2: The FNN when the fifth phase of the proposed approach is ignored and instead just one feature with the highest final weight in each cluster together with significant features in **TraRefWei1&2** which do not belong to any cluster are selected. The FNN is trained using these features.

(7) UpperBound1: The FNN trained by labeled instances in the target domain using all the features of the target domain.

(8) UpperBound2: The FNN trained by labeled instances in the target domain using features of the target domain which are selected by the proposed approach. This is expected to be the best baseline for evaluating the other baselines.

(9) UpperBound3: Thee FNN trained by labeled instances in the target domain using features selected in the baseline **TraRefWei1R2.**

5.4.2.2 Experiment Design using Real World Financial Data Sets

The performance of three baselines, NoTra; TraRefWei1&2 and UpperBound2, explained in the previous section, are benchmarked with those of two existing domain adaptation approaches. Since most existing methods assume that the source and target

domains are defined by the same features, they cannot be directly applied to these experiments. Few studies investigate the situation in which domains have different feature spaces, which is called heterogeneous domain adaptation. We apply two recent efficient heterogeneous domain adaptation approaches for comparison in this section.

(1) Manifold Alignment using Correspondences (MAC)(Wang & Mahadevan 2009): The key idea of this approach is to project different domains in a latent space, match the corresponding instances and preserve the topology of each input domain. It applies labeled and unlabeled data for domain adaptation and assumes that there are a limited number of labeled data in the target domain. Applying manifold alignment to domain adaptation in this approach needs to specify cross-domain correspondence relationships to learn the mapping function, which may be difficult to gain in most domain adaptation applications.

(2) Manifold Alignment using Labels (MAL)(Wang & Mahadevan 2011): This approach, which is an extension of the previous approach, explores how to use label information rather than correspondence to align input domains. The key idea underlying this approach is that many source and target domains defined by different feature spaces often share the same labels. Accordingly, it learns map functions to project the source and target domains to a new latent space, matches the instances of two domains with the same labels and preserves the topology of each input domain.

In each experiment, one State is selected as the Target State and one State, or a combination of States, is regarded as the Source State. There are $\sum_{k=1}^{11}\binom{11}{k} = 2^{11}$ combinations of 11 States as the Source State for each Target State. In total there are $12\sum_{k=1}^{11}\binom{11}{k} = 12 \times 2^{11}$ experiments to perform for each baseline. To simplify the results analysis and achieve a comprehensive conclusion we only compute the average of predictive accuracy for 12 combinations of $\binom{11}{1}$ and $\binom{11}{10}$. Experiment A is the combination $\binom{11}{1}$ denoting the experiments when there is one Source Sate for each Target State, and Experiment B is the combination $\binom{11}{10}$ representing the experiments when there is combination of 11 Source States for each Target State.

5.4.3 EXPERIMENT RESULTS ANALYSIS

This section reports the results of the experiments for the fuzzy cross-domain adaptation approach. Empirical analyses and comparisons show that the approach reduces predictive error and makes significant improvement.

5.4.3.1 EXPERIMENT RESULTS ANALYSIS USING SYNTHETIC DATA SETS

The approach is evaluated according to various baselines to explore the importance of each phase in the performance of the proposed approach. These baselines consider different situations ranging from worst case: training FNN using source data applied on unlabeled target data, to best case: training FNN using labeled target data. Applying different baselines assists with finding out how different phases and algorithms can influence on performance of approach. Figure 5.13 depicts the accuracy of different baselines using the synthetic data when FNN is applied as predictor.

Table 5.5 shows the maximum weight of the linguistic terms of each feature in the target domain which is computed by baselines. Features 1, 2 and 3, which belong to $DIFS$, are selected in the NoTra and TraRef baselines. NoTra, which is FNN trained by source data, has the worst predictive accuracy, while TraRef, which applies the MSFBR algorithm to refine the labels, achieves more than 5% improvement. TraRefWei1 is the baseline which selects Features 1 to 8 based on their primary weight ω_1.

Each of the three clusters, which are created after applying FSFA, has one core member of Features 1, 2 and 3. The cluster members are as follows:

$C_1 = \{Feature\ 1, Feature\ 4, Feature\ 5\}$

$C_2 = \{Feature\ 2, Feature\ 6, Feature\ 8\}$

$C_3 = \{Feature\ 3, Feature\ 7\}$

TraRefWei1 significantly gains more predictive accuracy than previous baselines as a result of applying the FSFA algorithm to select eight relevant features based on their correlation to the features of $DIFS$. The improvement, which is about 10% compared to TraRef, demonstrates the role of the FSFA algorithm on increasing the accuracy. TraRefwei2 selects nine features according to their secondary weights ω_2. Although the number of features is greater than of the number of features used by TraRefWei1, TraRefwei2 has less accuracy than TraRefWei1. The decrease in performance by 5% may be the result of some important missing features that have a significant influence on prediction. Although more significant features, not correlated to the DIFS features, are selected, other important features with high correlation to DIFS features are not nominated for prediction.

TraRefWei_1&2 uses the combination of primary and secondary weights to select 10 significant features. It shows more than 6% improvement in predictive accuracy compared to TraRefWei1. The reason is that it applies the efficient union of features which are used in TraRefWei1 and TraRefWei2. TraRefWei_1R2 is an interesting baseline which uses few features to achieve high accuracy. As can be seen in Table 5.5, it selects five features for prediction. The first three features (Features 3, 4 and 8) are representative of each cluster with the highest final weight, and the other features (Features 10 and 11) are significant features with a high final weight which are also selected in TraRefWei_1&2. Although it performs less accurately than TraRefWei_1&2 by 2%, TraRefWei_1R2 uses half the number of features that TraRefWei_1&2 uses for prediction. It can be concluded that TraRefWei_1R2 can be employed in data sets with a large number of features more efficiently than TraRefWei_1&2 in terms of computation time. However, it may perform less accurately than TraRefWei_1&2.

UpperBound baselines are the FNN which are trained using grand truth labels in the target domain. UpperBound1, which considers all features, obtains less accuracy than other UpperBound baselines. UpperBound 2 and 3 use features that are selected in TraRefWei_1&2 and TraRefWei_1R2 respectively and achieve superior performance

to UpperBound 1. Although both baselines have close performance, UpperBound 2 outperforms UpperBound 3 by 1%. Since these FNNs are trained on the target domain and the error caused by the distribution difference between domains is disregarded, the superior performance of UpperBound 2 and 3 to that of UpperBound 1 by 5% implies that a feature selection process using source domain knowledge can greatly improve prediction model accuracy in the target domain.

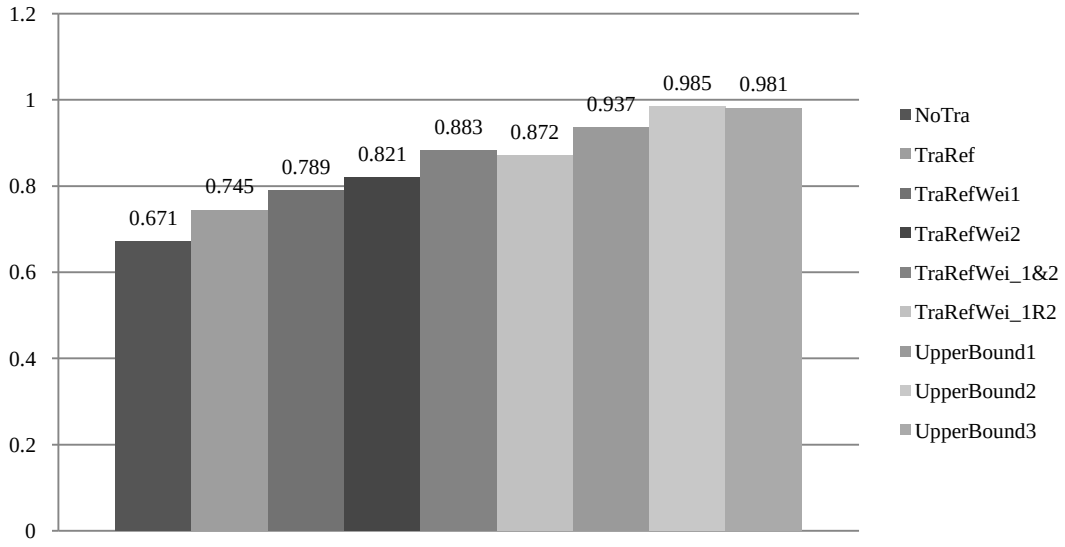


FIGURE 5.13: ACCURACY (GM) OF DIFFERENT BASELINES

TABLE 5.4: FEATURE WEIGHT AND BASELINES ACCURACY (GM)

	NoTra	TraRef	TraRefWei1	TraRefWei2	TraRefWei 1&2	TraRefWei 1R2	Upper Bound1	Upper Bound2	Upper Bound3
Feature 1	1	1	1	0.846	0.884	0.884	1	0.884	0.884
Feature 2	1	1	1	0.567	0.675	0.900	1	0.675	0.900
Feature 3	1	1	1	0.912	0.934	0.934	1	0.934	0.934
Feature 4	0	0	0.893	0.873	0.878	0.878	1	0.878	0.878
Feature 5	0	0	0.712	0.804	0.781	0.781	1	0.781	0.781
Feature 6	0	0	0.818	0.629	0.676	0.629	1	0.676	0.629
Feature 7	0	0	0.718	0.732	0.728	0.728	1	0.728	0.728
Feature 8	0	0	0.805	0.937	0.904	0.904	1	0.904	0.904
Feature 9	0	0	0.245	0.581	0.497	0.497	1	0.497	0.497
Feature 10	0	0	0.068	0.889	0.684	0.684	1	0.684	0.684
Feature 11	0	0	0.214	0.849	0.690	0.690	1	0.690	0.690
Feature 12	0	0	0.009	0.825	0.621	0.621	1	0.621	0.621
Feature 13	0	0	0.129	0.157	0.150	0.150	1	0.150	0.150
Feature 14	0	0	0.315	0.308	0.309	0.309	1	0.309	0.309
Feature 15	0	0	0.148	0.212	0.196	0.196	1	0.196	0.196
N.O features	4	4	8	9	10	5	15	10	5
Accuracy	0.671	0.725	0.821	0.789	0.883	0.862	0.937	0.985	0.971

5.4.3.2 EXPERIMENT RESULTS ANALYSIS USING REAL WORLD FINANCIAL DATA SETS

In this section, three main baselines NoTra, TraRefWei1&2 and UpperBound2, which have been explained in a previous section, are benchmarked against two heterogeneous domain adaptation approaches. The reason for choosing these baselines is that they achieved the best performance in previous experiments, particularly when the problem included a small number of features. There are few features in the banking data set, as explained in Section 5.4.1.2. The experiments are carried out according to the experiment design described in Section 5.4.2.2. The results of these experiments are demonstrated in Tables 5.6, 5.7 and Figures 5.14 to 5.19. The final weights of features when TraWei1&2 is applied are shown in Tables 5.8 and 5.9. Table 5.10 shows the statistical tests which have been conducted to compare the performance of the TraRefWei1&2 approach with other approaches. The

performance of TraRefWei1&2 in Experiments A and B are benchmarked against each other using statistical tests and the results are showed in Table 5.11.

Experiment A:

This experiment is carried out to evaluate the performance of approaches when only one source domain is available for transfer learning. Each experiment is performed considering one State as the target domain and one State as the source domain. There are 11 different source domains and consequently 11 different experiments for each target State. Each item in Table 6 indicates the average accuracy (GM) of these 11 different experiments for each State and each approach. The approaches are tested for three year of predictions: same year (2000), two years ahead (1998) and five years ahead (1995). According to the results, the accuracy of all approaches decreases when the period of prediction becomes longer. The average accuracy of the proposed approach (TraRefWei1&2) is different for each State; however, it does not follow any pattern related to the data set. For instance, some States have less data than other States, yet they outperform them. Moreover, any relation between the degree of imbalance problem in the data set and the average accuracy cannot be found. The average accuracy of different approaches for each State and each period of prediction is depicted in Figures 5.14, 5.15 and 5.16 using the data in Table 5.6. These results, gained from a single source domain, clearly shows that although UpperBound 2 has the best accuracy, as would be expected, the proposed approach outperforms all other approaches in all experiments. To examine this improvement more deeply, the Holm test (Holm 1979) is performed using all the accuracy values achieved in Experiment A for each approach using a 0.05 level of significance. The results, which are illustrated in Table 5.10, conclude that the proposed approach significantly outperforms other approaches with 95% of confidence. The hypothesis of equality of TraRefWei1&2 and MAC accuracy is only Not Rejected when the period of prediction reaches five years. Although UpperBound 2 has superior performance compared to the proposed approach, it is not significant according to the Holm test, and their accuracies are equal with 95% of confidence.

TABLE 5.5: AVERAGE ACCURACY IN EXPERIMENT A

Baselines	Year of Prediction	Target States											
		IL	TX	MN	IA	CA	NY	KS	MO	GA	WI	FL	OK
NoTra	2000	88.414	87.209	88.915	88.171	87.909	89.349	87.723	87.518	88.291	89.534	88.878	87.149
	1998	83.157	83.567	84.156	82.881	83.478	83.252	82.777	83.714	83.019	84.315	83.492	83.741
	1995	79.512	80.535	79.533	79.212	80.157	80.300	80.453	79.893	78.692	79.985	80.816	79.566
MAC	2000	91.303	91.159	90.512	90.492	91.537	89.924	90.654	90.451	90.370	91.398	91.252	90.879
	1998	88.568	88.247	87.383	88.172	87.814	87.412	87.062	88.351	87.509	88.621	87.205	87.710
	1995	83.112	83.670	83.627	82.644	83.207	82.809	83.799	83.449	82.960	83.381	83.931	83.828
MAB	2000	92.421	93.084	92.014	92.369	92.715	91.712	92.396	92.412	92.303	91.919	92.184	92.279
	1998	88.077	89.055	89.669	89.470	89.202	89.856	89.365	89.534	89.301	89.477	90.008	88.737
	1995	85.303	86.720	85.196	85.039	85.881	86.360	86.034	85.450	86.198	85.086	85.503	85.646
TraRefWei1&2	2000	95.552	95.158	95.719	94.612	95.273	95.836	95.980	95.071	95.215	94.719	95.291	95.314
	1998	91.739	91.802	91.545	92.262	91.852	91.441	91.358	91.635	91.517	91.118	91.316	89.839
	1995	87.187	87.109	87.362	86.924	87.394	87.555	87.516	87.781	87.291	88.196	87.651	87.718
UpperBound2	2000	97.412	98.661	98.125	98.412	98.789	98.367	98.115	97.649	98.915	98.225	98.618	98.357
	1998	92.379	93.114	93.395	93.532	92.925	93.760	93.650	93.952	93.469	92.772	94.197	93.121
	1995	87.058	88.844	89.222	89.559	89.458	89.506	89.375	90.016	88.965	89.932	89.327	89.746

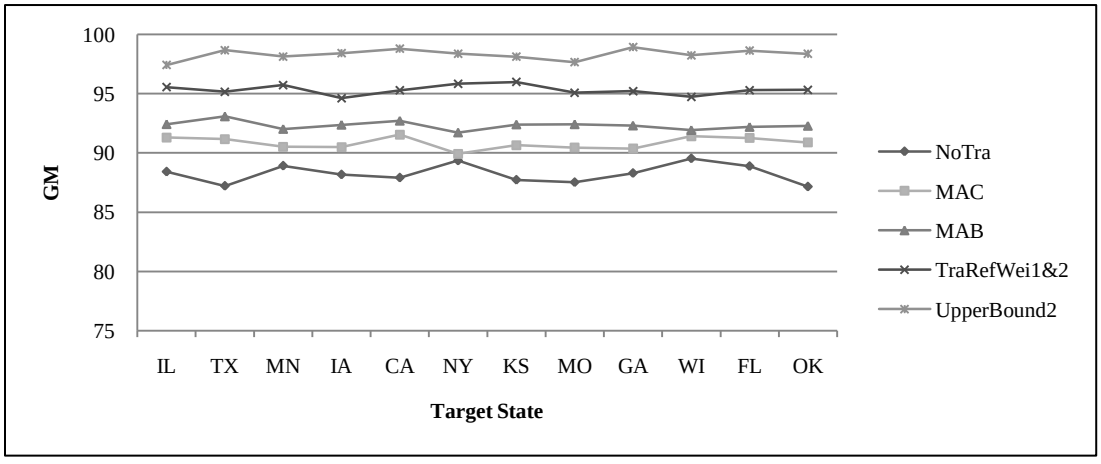


FIGURE 5.14: ACCURACY OF PREDICTION FOR YEAR 2000 IN EXPERIMENT A

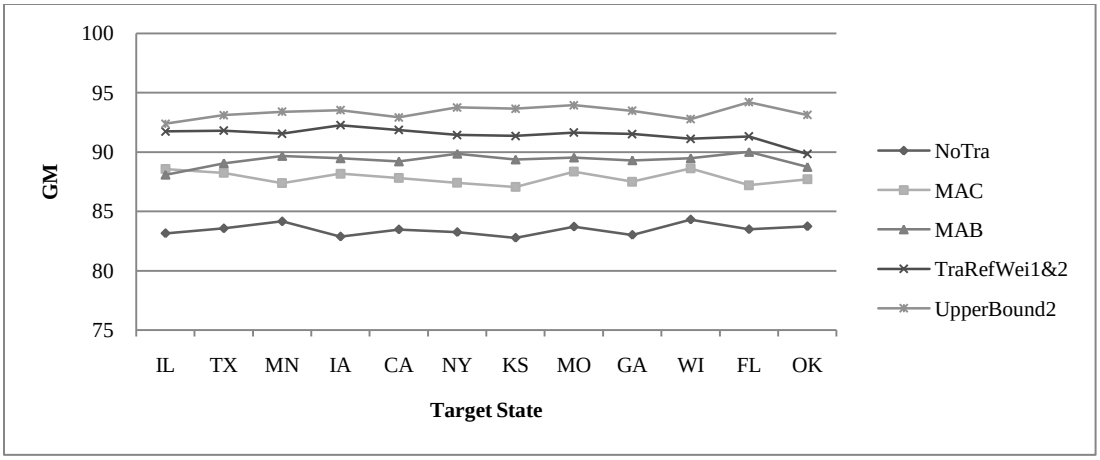


FIGURE 5.15: ACCURACY OF PREDICTION FOR YEAR 1998 IN EXPERIMENT A

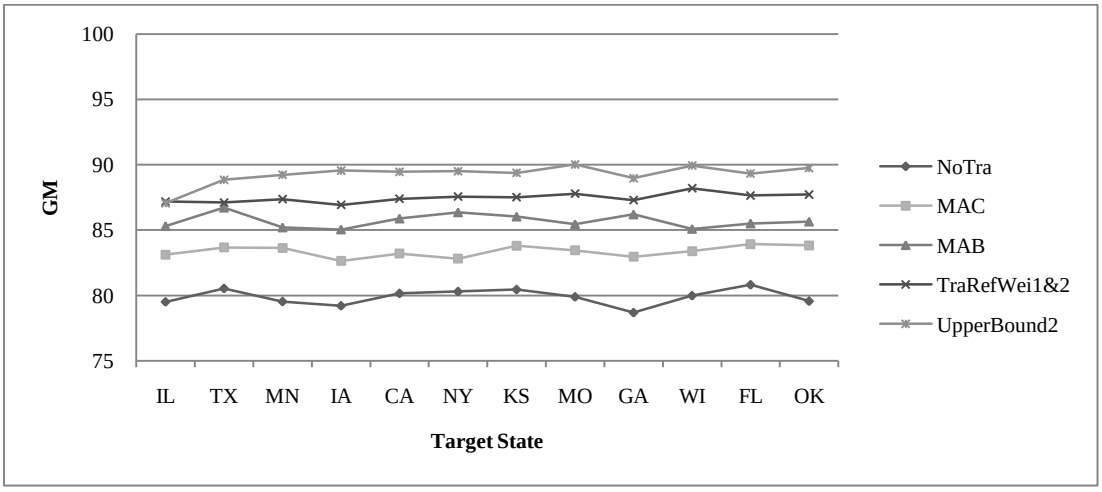


FIGURE 5.16: ACCURACY OF PREDICTION FOR YEAR 1995 IN EXPERIMENT A

Experiment B:

This experiment is performed to evaluate the approaches' performance when multiple source domains are available for transfer learning. Each experiment is carried out using one State as the target domain and 10 out of 11 other States as the source domains. There are 11 combinations of 10 States and consequently 11 experiments for each target State. Each cell in Table 5.7 computes the average of the accuracy values (GM) achieved from these experiments for each State and each approach. Similar to Experiment A, the average accuracy declines as the period of prediction lengthens. Moreover, the accuracy among various States changes and there is no pattern related to the data set that can explain the differences. According to the values

of Table 5.7, Figures 5.17, 5.18 and 5.19 are created to benchmark the performance of baselines for prediction in years 2000, 1998 and 1995 respectively. As the charts demonstrate, the UpperBound 2 and TraRefWei1&2 gain the best and second best rank respectively among the baselines for all target States. To examine these comparisons accurately, the Holm test, which is a nonparametric statistical test, is performed for all States in each period of prediction. The results, which are presented in Table 5.10, show that the hypothesis of equality of TraRefWei1&2 accuracy with that of other approaches is rejected with 95% of confidence and accordingly the proposed approach significantly improves the predictive accuracy. The difference between the performance of the proposed approach and that of UpperBound 2, which is trained by labeled target domain instances, is not significant.

The proposed approach calculates the final weight for each feature in the target domain and according to the predefined threshold (ε) in Phase Five, some of the features selected for training. The average of these final weights in each target State using 11 different experiments is computed. Tables 5.8 and 5.9 show these values in experiments A and B respectively. The number of selected features in every target State based on value $\varepsilon = 0.65$ are also indicated. The optimum value for the predefined threshold is achieved experimentally. As it can be seen, the features' final weights and accordingly selected features change among States. For instance, TraRefWei1&2 selects all 9 features for State *CA* while it nominates 5 features for States *MO* and *FL* in Experiment A. Similarly, selected features are different among target States to obtain the maximum accuracy in Experiment B; however, all target States apply the three first features belonging to $DIFS$ for transfer learning, because they gain high final weights due to their primary weights. Another predefined parameter is the trade-off of parameter β to combine the primary and secondary weight and compute the final weight. The value of $\beta = 0.7$ is experimentally computed.

TABLE 5.6: AVERAGE ACCURACY IN EXPERIMENT B

Experiment B		Target States											
Baselines	Year of Prediction	IL	TX	MN	IA	CA	NY	KS	MO	GA	WI	FL	OK
	2000	88.310	89.182	88.117	89.176	88.481	88.583	88.788	89.280	88.492	88.278	**88.91**	88.295
NoTra	1998	83.457	83.235	83.277	84.198	83.805	83.549	84.212	83.725	82.505	83.514	83.287	83.416
	1995	79.440	79.223	79.890	79.504	79.669	79.088	80.211	80.162	80.159	79.529	80.195	80.591
	2000	91.038	90.312	90.667	90.853	90.559	91.148	91.228	91.316	90.781	90.470	91.663	90.975
MAC	1998	87.925	88.759	88.343	87.946	87.435	87.637	88.026	87.754	87.532	87.771	88.508	87.918
	1995	83.290	83.633	83.670	84.410	82.898	83.690	83.403	84.116	84.048	83.752	**83.41**	82.328
	2000	93.020	93.040	92.853	92.620	92.861	92.794	93.019	92.933	92.450	93.483	92.535	92.674
MAB	1998	89.361	89.403	89.384	88.975	89.432	89.111	89.330	90.124	89.809	90.159	90.148	89.395
	1995	85.327	86.308	85.803	86.092	85.978	86.274	87.041	86.432	85.453	86.532	86.096	86.604
	2000	95.796	95.600	95.860	96.384	96.513	96.090	96.381	95.486	96.654	95.965	96.726	96.348
TraRefWei1&2	1998	92.284	92.347	93.212	93.167	92.810	92.568	93.015	92.649	92.354	92.303	92.494	92.224
	1995	88.799	87.947	88.561	89.185	89.238	87.492	88.883	89.048	89.161	89.061	88.332	89.078
	2000	98.488	98.475	98.466	98.903	98.384	98.851	98.357	99.199	98.593	99.182	99.515	98.725
UpperBound 2	1998	93.387	93.538	94.138	93.686	93.752	93.506	93.439	93.660	94.053	93.692	93.671	93.843
	1995	89.020	89.274	89.515	89.798	90.745	90.180	90.522	89.799	89.626	89.368	91.033	90.169

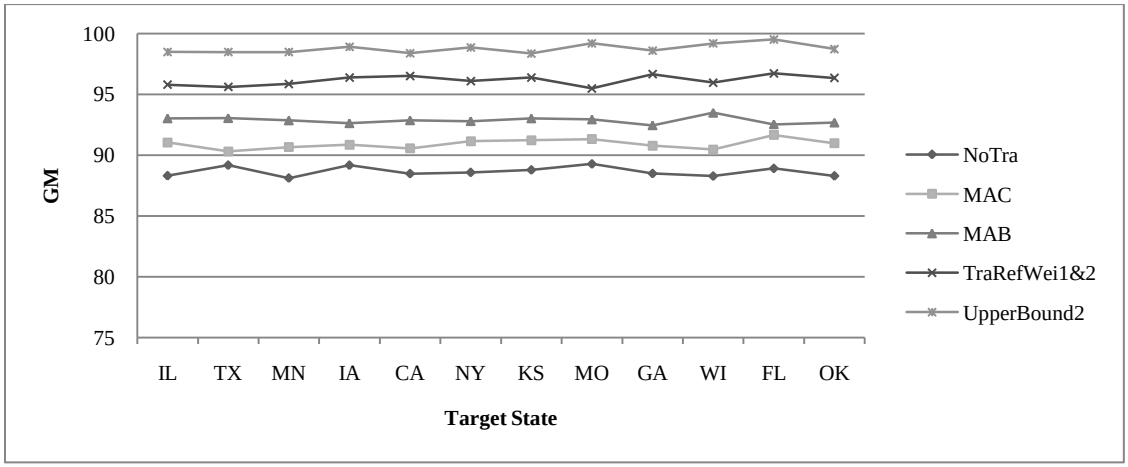


FIGURE 5.17: ACCURACY OF PREDICTION FOR YEAR 1998 IN EXPERIMENT B

FIGURE 5.18 : ACCURACY OF PREDICTION FOR YEAR 1998 IN EXPERIMENT B

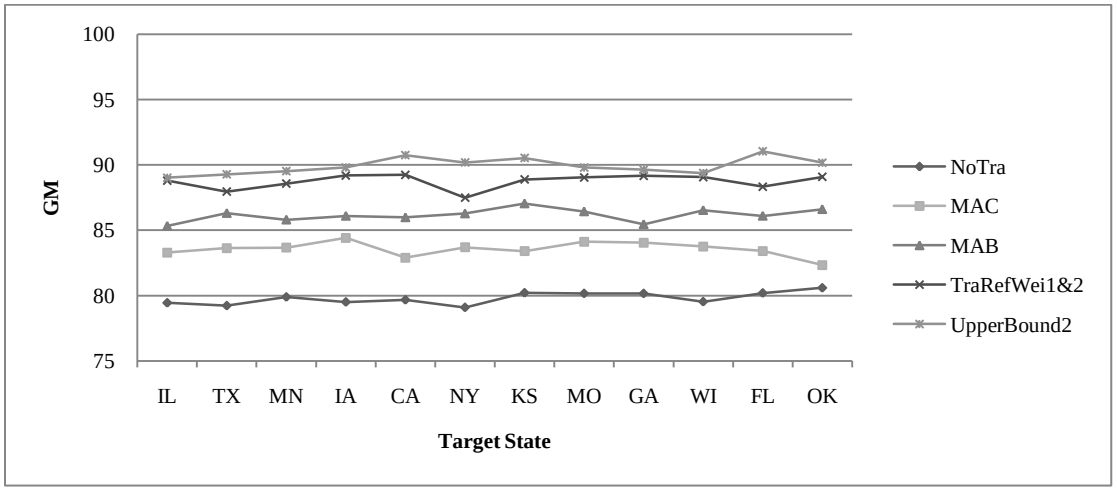


FIGURE 5.19: ACCURACY OF PREDICTION FOR YEAR 1995 IN EXPERIMENT B

The Holm test is also applied to accurately compare the performance of the proposed approaches in Experiments A and B. Table 5.11 demonstrates the results of the statistical tests in which the hypothesis of equality of accuracy is rejected in favor of Experiment B with 95% of confidence. It concludes that the performance of the proposed approach improves significantly when multiple source domains are applied for transfer learning.

TABLE 5.7: FINAL FEATURE WEIGHTS FOR TRAWEI1&2 IN EXPERIMENT A

	IL	TX	MN	IA	CA	NY	KS	MO	GA	WI	FL
Feature1	0.85	0.76	0.88	0.84	0.91	0.82	0.82	0.96	0.83	0.84	0.93
Feature2	0.81	0.75	0.84	0.82	0.86	0.80	0.90	0.87	0.82	0.94	0.90
Feature3	0.79	0.84	0.94	0.78	0.89	0.73	0.85	0.67	0.87	0.78	0.79
Feature4	0.77	0.89	0.87	0.58	0.69	0.51	0.82	0.90	0.66	0.61	0.61
Feature5	0.89	0.59	0.78	0.87	0.73	0.78	0.58	0.64	0.67	0.71	0.58
Feature6	0.58	0.62	0.68	0.59	0.65	0.82	0.67	0.54	0.73	0.88	0.87
Feature7	0.88	0.79	0.60	0.61	0.71	0.74	0.79	0.81	0.59	0.75	0.60
Feature8	0.84	0.83	0.61	0.75	0.68	0.58	0.84	0.63	0.70	0.67	0.72
Feature9	0.73	0.84	0.81	0.89	0.66	0.87	0.91	0.61	0.69	0.59	0.57
N.O Features	8	7	6	6	9	7	8	5	8	7	5

TABLE 5.8: FINAL FEATURE WEIGHTS FOR TRAWEI1&2 IN EXPERIMENT B

	IL	TX	MN	IA	CA	NY	KS	MO	GA	WI	FL
Feature1	0.86	0.83	0.86	0.91	0.79	0.93	0.84	0.92	0.87	0.85	0.84
Feature2	0.79	0.83	0.89	0.82	0.86	0.89	0.84	0.85	0.89	0.84	0.93
Feature3	0.92	0.80	0.91	0.85	0.86	0.88	0.90	0.87	0.82	0.81	0.93
Feature4	0.82	0.76	0.85	0.68	0.59	0.83	0.91	0.59	0.62	0.61	0.79
Feature5	0.84	0.71	0.76	0.80	0.63	0.62	0.66	0.76	0.86	0.75	0.90
Feature6	0.93	0.59	0.76	0.73	0.84	0.73	0.56	0.80	0.62	0.79	0.86
Feature7	0.57	0.63	0.72	0.59	0.74	0.84	0.68	0.59	0.89	0.72	0.81
Feature8	0.85	0.91	0.64	0.57	0.64	0.53	0.59	0.79	0.59	0.69	0.92
Feature9	0.83	0.90	0.82	0.89	0.61	0.85	0.79	0.59	0.84	0.75	0.57
N.O Features	8	7	8	7	5	7	7	6	6	8	8

TABLE 5.9: HOLM TEST FOR COMPARISON OF TRAREFWEI1&2 WITH OTHER BASELINES

Experiment	Hypothesis	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Result
A-2000	TraRefWei1&2 vs. NoTra	4.648	3.358E-6	0.006	Rejected
	TraRefWei1&2 vs. MAC	3.098	0.002	0.008	Rejected
	TraRefWei1&2 vs. MAB	3.098	0.002	0.010	Rejected
	TraRefWei1&2 vs. UpperBound 2	1.549	0.121	0.050	Not Rejected
A-1998	TraRefWei1&2 vs. NoTra	4.648	3.358E-6	0.006	Rejected
	TraRefWei1&2 vs. MAC	3.227	0.001	0.007	Rejected
	TraRefWei1&2 vs. MAB	2.969	0.002	0.010	Rejected
	TraRefWei1&2 vs. UpperBound 2	1.549	0.121	0.025	Not Rejected
A-1995	TraRefWei1&2 vs. NoTra	4.776	1.782E-6	0.006	Rejected
	TraRefWei1&2 vs. MAC	3.227	0.001	0.007	Rejected
	TraRefWei1&2 vs. MAB	1.678	0.093	0.012	Not Rejected
	TraRefWei1&2 vs. UpperBound2	1.290	0.197	0.050	Not Rejected
B-2000 & 1998 & 1995	TraRefWei1&2 vs. NoTra	4.648	3.358E-6	0.006	Rejected
	TraRefWei1&2 vs. MAC	3.098	0.002	0.008	Rejected
	TraRefWei1&2 vs. MAB	3.098	0.002	0.010	Rejected
	TraRefWei1&2 vs. UpperBound 2	1.549	0.121	0.050	Not Rejected

TABLE 5.10: HOLM TEST FOR COMPARISON OF TRAREFWEI1&2 IN EXPERIMENTS A AND B

Hypothesis	Year of Prediction	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Result
Single Source (Experiment A) vs. Multiple Source (Experiment B)	2000	3.464	5.320E-4	0.05	Rejected
	1998	3.464	5.320E-4	0.05	Rejected
	1995	2.886	0.004	0.05	Rejected

5.5 SUMMARY

In this chapter, a fuzzy cross-domain adaptation approach is proposed to solve the domain adaptation problem in which the feature space of the target domain is different from that of the source domain. This approach concentrates on finding significant features in the discriminative task of the target domain using the knowledge of the source domain which has similar features to the target domain. The fuzzy neural network is employed on domain-independents feature space as a prediction model to determine the initial labels for target instances (Phase 1). The

MSFBR algorithm is applied to modify the instances' labels in the target domain (Phase 2). In particular, this study develops a FSFA algorithm, which is the main part of the fuzzy cross-domain adaptation approach and the major contribution of this research, to explicitly depict the relation between two domains and explore the correlated fuzzy features of both domains. As a result of performing FSFA, the most highly-correlated features in the target domain are weighted (Phase 3). According the refined labels in Phase 2, the significant features are weighted by a fuzzy genetic feature weighting algorithm (Phase 4). Finally, by combining the results of Phases Three and Four, the most significant features in the target domain are selected and the model is retrained (Phase 5). The proposed approach is validated and compared with existing approaches using synthetic and real financial data for bank failure prediction. The empirical results show that the proposed algorithm successfully explores the significant fuzzy features in the target domain using the common knowledge of the source domains. It demonstrates a significant increase in predictive accuracy, particularly when the algorithm utilizes multiple source domains for training (Experiment B). The results show that the proposed approach offers even better enhancement when it is applied to a greater time period prediction. Compared to other methods which apply crisp-value, this approach applies a fuzzy concept to modify the predicted labels in Phase Two and find the significant features in the target domain in Phases 3 and 4. Consequently, it achieves better results. The proposed approach is more general and relatively independent from the predictive function and can be applied with other prediction methods. It can successfully transfer knowledge over different domains and a long time period to predict bank failure ten years ahead. The approach can be considered as an applicable prediction model which does not need to be re-trained in every determined domain and/or period.

CHAPTER 6

CASE STUDY: AUSTRALIAN BANKS EXPERIENCE

6.1 INTRODUCTION

In this chapter, we focus on solving the transductive transfer learning problem in which the source domain is the United States banking system and the target domain is the Australian banking system. This chapter considers two main approaches according to two categories of domain adaptation. The first approach assumes that the significant features (financial ratios) for failure prediction are known and are the same in both domains, but that the marginal distribution of data is different. The goal is to exploit the plentiful labeled data in the source domain to predict the label for target instances. The second approach goes further and assumes that only the significant features in the source domain are known and that a few of them are common in both domains. We aim to find the significant features for bank failure prediction in the target domain using the knowledge available in the source domain and then predict labels for the target instances.

This chapter is organized as follows: Section 6.2 introduces the setting of the problem we aim to solve in this chapter. Section 6.3 outlines the modeling, including the algorithms for the first approach, the fuzzy domain adaptation approach, and the second approach, cross domain adaptation. Section 6.4 presents the evaluation and

analysis of the experimental results for bank failure prediction. Finally, the summary of the chapter is discussed in Section 6.5.

6.2 PROBLEM SETTING AND DEFINITIONS

In this section, the domain adaptation problem in bank failure, which this study intends to solve, is explained in detail and the notations to be used throughout the chapter are introduced. The problem settings are explained on the basis of two main assumptions. These assumptions correspond to two categories of domain adaptation problems and are adapted for the bank failure prediction problem which we aim to solve. We aim to predict the financial health of Australian banks (target domain) and assign a binary label as failed or survived to each bank using the labeled United States banks (source domain). Hence, the source domain and target domain are referred to as the United States and Australian banking systems respectively and the task is a binary bank failure prediction and is the same for both domains. The definitions of source and target domains and the bank failure prediction task are presented as follows:

Definition 6.1 Source Domain **(United States banking system)** is denoted by $D_s = \{F_s, P_s(X)\}$, consists of two components:

(1) Feature space of source domain (United States banking system) $F_s = \{f_{s_1}, \dots, f_{s_{l_s}}\}$ where l_s is the number of features (significant financial ratios) in the source domain (United States banking system); and

(2) Marginal probability distribution of instances in the source domain (United States banks) $P_s(X_s)$, where $X_s = \left\{x_{s_1}, \dots, x_{s_{n_s}}\right\} \in F_s$ where n_s are the number of instances in the source domain (United States banks).

Definition 6.2 Target Domain **(Australian banking system)** is denoted by $D_t = \{F_t, P_t(X)\}$, consists of two components:

(1) Feature space of target domain (Australian banking system) $F_t = \{f_{t_1}, \dots, f_{t_{l_t}}\}$ where l_t is the number of features (significant financial ratios) in the target domain (Australian banking system); and

(2) Marginal probability distribution of instances in target domain (Australian banks) $P_t(X_t)$, where $X_t = \left\{x_{t_1}, \dots, x_{t_{n_t}}\right\} \in F_t$ where n_t are the number of instances in the target domain (Australian banks).

Definition 6.3 Task **(binary bank failure prediction)** is denoted by $T = \{Y, f(\cdot)\}$, consists of two components:

(1) A label space $Y = \{y_1, y_2\}$ where $y_1 = -1$ refers to survived banks (Negative class) and $y_2 = 1$ refers to failed banks (Positive class) ; and

(2) An objective predictive function (bank failure prediction model) $f(\cdot)$ which is not observed and to be learned by pairs $\{x_i, y_i\}$ in both domains.

In both problem settings, we assume that the problem is an unsupervised domain adaptation problem in which no labeled Australian banks (target domain) are available while many labeled United States banks (source domain) are available. In Setting One, we assume that both domains have the same feature spaces but the data have different marginal distributions. In Setting Two, we assume that the feature spaces are different but that there are universal features which are the same in both domains. Data marginal distributions are also different. According to these assumptions, which refer to the first and second category of domain adaptation problem, the following definitions of domain adaptation and cross-domain adaptation problems are supplied.

Definition 6.4 (Domain adaptation problem-Setting One) Given (1) United States banks data as source domain D_s; (2) The bank failure prediction model trained by data in the source domain (shift-unaware prediction model) as the source learning task T_s; (3) Australian banks data as a target domain D_t; and (4) The bank failure prediction model for instances in the target domain as the target learning task T_t, the domain adaptation approach aims to improve the learning of the target predictive function $f_t(\cdot)$ in D_t using the knowledge in D_s and T_s where the significant financial

ratios (feature spaces) between Australian and United States banking systems (domains) are the same ($F_t = F_s$), but the marginal probability distributions of banks are different in both systems ($P_t(x) \neq P_s(x)$).

Definition 6.5 (Cross-domain adaptation problem-Setting Two) Given (1) United States banks data as source domain D_s; (2) The bank failure prediction model trained by data in the source domain (shift-unaware prediction model) as the source learning task T_s; (3) Australian banks data as a target domain D_t; and (4) The bank failure prediction model for instances in the target domain as the target learning task T_t, the domain adaptation approach aims to improve the learning of the target predictive function $f_t(\cdot)$ in D_t using the knowledge in D_s and T_s where the significant financial ratios (feature spaces) between Australian and United States banking systems (domains) are the different (F), but there are a number of financial ratios which are common between two banking systems . Also, the marginal probability distributions of banks are different in both systems (P)).

6.3 MODELLING

In this section three types of domain adaptation algorithms called Multi-Step Fuzzy Bridge Refinement algorithms (MSFBR) Type I, II and III are proposed to solve the domain adaptation problem-Setting One. These algorithms will be benchmarked in the next section using real financial data. We first describe the related theory of the proposed algorithms and then present the MSFBR algorithms and their implementations based on the explained theory.

6.3.1 FUZZY BRIDGED REFINEMENT DOMAIN ADAPTATION (FIRST APPROACH)

Bridged Refinement theory, which is motivated by PageRank theory (Page et al. 1998), assumes that the conditional probability of a specified label C, given an instance d, does not vary between different distributions: P

although the marginal probability of instance d ($P(d)$) varies. This is based

on the fact that, if an identical instance appears in the target domain and the source domain, the predicted label should be the same. The more similar instances that are in the target domain, the greater the probability that they have the same label. This situation creates a mutual reinforcement relationship between instances in the two domains and can be used to correct the predicted labels. Not only is this assumption considered in this research, but a complementary idea is also applied. We assume that the more different the instances are in the target domain, the less probability there is that they will have the same label. For instance, in a two class problem, significantly dissimilar instances are located in opposite classes, while the significantly similar instances are located in the same class. In other words, the similarity and dissimilarity between instances simultaneously indicates their class labels. However, the similarity and dissimilarity functions play an important role and need to be defined well enough for mapping the instances and then discriminating the instances accurately. Recently Balcan (2008) and Wang (2007) developed theories for good similarity and dissimilarity functions and gave sufficient conditions for the functions to allow one learn well. Hence, the definitions and conditions can be used to define similarity and dissimilarity functions such that there is a high probability that similar instances will have the same labels and dissimilar instances will have different labels. We used the definition and theory (Definitions 6.1 and 6.2 and, Theorems 6.1 and 6.2 proposed by Wang et al. (2008; 2007) to define and construct our functions and the most similar and dissimilar instances to a target instance are then applied to modify the class label.

Let $x_i, x_j \in D$, where D is a given domain. The similarity and dissimilarity functions are denoted as follows:

- Similarity function: $Sm(x_i, x_j)$ where $x_i, x_j \in X$ and $Sm(x_i, x_j) \in [0, 1]$;
- Dissimilarity function: $Dm(x_i, x_j)$ where $x_i, x_j \in X$ and $Dm(x_i, x_j) \in [0, 1]$.

Definition 6.1 (Wang et al. 2007) Let $z_a, z_b, z_c \in D \times \{-1, 1\}$ are labeled instances in a given domain. Similarity (Sm) and dissimilarity (Dm) functions are strongly (ε, γ)-good for the learning problem if at least $1 - \varepsilon$ mass probability of instances z satisfy:

$p(Sm(x_a, x_b) > Sm(x_a, x_c) \,|y_a = y_b, y_a = -y_c) \geq 0.5 + \frac{\gamma}{2}$;

$p(Dm(x_a, x_b) < Dm(x_a, x_c) \,|y_a = y_b, y_a = -y_c) \geq 0.5 + \gamma/2,$

where the probability is over random instances z_b, z_c.

This definition says that Sm (respectively, Dm) is a strongly good similarity (respectively, dissimilarity) function for a learning problem if most instances (at least $1 - \varepsilon$ mass probability) are on average at least γ more similar (respectively, dissimilar) to random instances $x_b(x_c)$ of the same (respectively, opposite) label than they are to random instances x_c (x_b) of the opposite (respectively, same) label.

Theorem 6.1 (Wang et al. 2007) If *Sm* and *Dm* are strongly (ε, γ)- good, then $\left(\frac{4}{\gamma^2}\right) \ln\left(\frac{2}{\delta}\right)$ positive examples and $\left(\frac{4}{\gamma^2}\right) \ln\left(\frac{2}{\delta}\right)$ are sufficient so that with the *probability* $\geq 1 - \delta$, the above algorithm produces a classifier $U(x)$ with error at most $\varepsilon + \delta$.

Proof: See (Wang et al. 2007).

The theory suggests that by using a sufficiently large set of positive and negative instances and similarity or dissimilarity functions, the constructed classifier will specify the label of given instances accurately (error $\leq \varepsilon + \delta$). The following definition introduces a less strict definition for good similarity and dissimilarity functions. Next, Theorem 6.2 presents a classifier which is formed by the introduced similarity/dissimilarity functions.

Definition 6.2 (Wang et al. 2007) Let $z_a, z_b, z_c \in D \times \{-1,1\}$, similarity *Sm* and dissimilarity *Dm* functions are $(\varepsilon, \gamma, B, \pi)$- good for learning problem if:

There are two conditional pdfs such that $\frac{\tilde{p}(x|y=1)}{p(x|y=1)} \leq \sqrt{B}$, $\frac{\tilde{p}(x|y=-1)}{p(x|y=-1)} \leq \sqrt{B}$ hold for at least $1 - \pi$ mass probability of instances z.

There is a threshold $v(x_b, x_c)$ such that at least $1 - \varepsilon$ probability mass of examples z satisfy:

$\tilde{p}(Sm(x_a, x_b) - Sm(x_a, x_c) > v(x_b, x_c) \,|y_a = y_b, y_a = -y_c) \geq 0.5 + \gamma/2$

or

$\tilde{p}(Dm(x_a, x_b) - Dm(x_a, x_c) < v(x_b, x_c) \,|y_a = y_b, y_a = -y_c) \geq 0.5 + \gamma/2.$

Theorem 6.2 (Wang et al. 2007) If Sm and Dm are $(\varepsilon, \gamma, B, \pi)$- good functions**,** then $n = \left(\frac{16B^2}{\gamma^2}\right) \ln\left(\frac{2}{\delta}\right)$ positive x_{b_i} and $n = \left(\frac{16B^2}{\gamma^2}\right) \ln\left(\frac{2}{\delta}\right)$ negative x_{c_i} instances are sufficient so that with the *probability* $\geq 1 - \delta$, there exists a convex combination classifier $g(x)$ of *n* base classifiers $h_i(x_a)$:

$g(x_a) = \sum_{i=1}^{n} \omega_i \, h_i(x_a), \;\; \sum \omega_i = 1, \; \omega_i \geq 0,$

where

$$h_i(x_a) = sgn\left[Sm\left(x_a, x_{b_i}\right) - Sm\left(x_a, x_{c_i}\right) + v\left(x_{b_i}, x_{c_i}\right)\right],$$

or

$$h_i(x_a) = sgn\left[Dm\left(x_a, x_{c_i}\right) - Dm\left(x_a, x_{b_i}\right) + v\left(x_{b_i}, x_{c_i}\right)\right].$$

such that the error rate of the combined classifier at margin $\gamma/2B$ is at most $\varepsilon + \delta$, provided $\pi \leq \frac{\gamma^2 \delta}{64B^2 \ln\left(\frac{2}{\delta}\right)}$ and the threshold is known.

Proof: See (Wang et al. 2007).

Using the classifier defined in Definition 6.2, we introduce a classifier based on the fuzzy concept. The labeled instances are presented by $z = (x, y) \in X \times Y$, where $x \in X$ is in fuzzy sets $\tilde{F}_k, k = 1, \ldots, l$ and $y \in Y$ is in fuzzy sets $\tilde{y}_k, k = 1,2$. $\mu_{\tilde{F}_k}(x)$ is the membership value of instance z in fuzzy set $\tilde{F}_k$ and $\mu_{\tilde{y}_1}(y)$ and $\mu_{\tilde{y}_2}(y)$are the membership values of instance z in negative (-1) and positive (1) classes respectively.

Definition 6.3: Let $z_a, z_{b_i}, z_{c_i} \in X \times Y \subseteq D$ where D is a given domain, $n = \left(\frac{16B^2}{\gamma^2}\right) \ln\left(\frac{2}{\delta}\right)$, *Sm* and *Dm* are $(\varepsilon, \gamma, B, \pi)$- good similarity and dissimilarity functions. $\forall z_a$, the classifier $U_D(x_a) = \left\{U_D^{\tilde{y}_1}(x_a), U_D^{\tilde{y}_2}(x_a)\right\} = \left\{\mu_{\tilde{y}_1}(y_a), \mu_{\tilde{y}_2}(y_a)\right\}$ which is constructed using instances of domain D, is defined as follows:

$$U_D^{\tilde{y}_1}(x_a) = \begin{cases} \left[\left(\frac{1}{n}\right) \sum_i^n \mu_{\tilde{y}_1}\left(y_{b_i}\right)\right] & \text{if} \quad g(x_a) \geq 0 \\ 1 - \left[\left(\frac{1}{n}\right) \sum_i^n \mu_{\tilde{y}_2}\left(y_{c_i}\right)\right] & \text{if} \quad g(x_a) < 0 \end{cases}$$

and

$$U_D^{\tilde{y}_2}(x_a) = \begin{cases} 1 - \left[\left(\frac{1}{n}\right) \sum_i^n \mu_{\tilde{y}_1}\left(y_{b_i}\right)\right] & \text{if} \quad g(x_a) \geq 0 \\ \left[\left(\frac{1}{n}\right) \sum_i^n \mu_{\tilde{y}_2}\left(y_{c_i}\right)\right] & \text{if} \quad g(x_a) < 0 \end{cases},$$

where

$$g(x_a) = \sum_{i=1}^{n} \omega_i \, h_i(x_a), \;\; \sum \omega_i = 1, \; \omega_i \geq 0$$

and

$$h_i(x) = sgn\left[Sm\left(x_a, x_{b_i}\right) - Sm\left(x_a, x_{c_i}\right) + v\left(x_{b_i}, x_{c_i}\right)\right]$$

or

$$h_i(x) = sgn\left[Dm\left(x_a, x_{c_i}\right) - Dm\left(x_a, x_{b_i}\right) + v\left(x_{b_i}, x_{c_i}\right)\right] \cdot$$

$U_D^{\tilde{y}_1}(x_a)$ and $U_D^{\tilde{y}_2}(x_a)$ indicate the membership values of the instance z_a in negative and positive classes respectively. The proposed classifier $U(\cdot)$ returns the mean of membership values of positive (negative) instances in the positive (negative) class based on the output of the $g(\cdot)$. According to the Definitions 6.2 and 6.3, and using the fuzzy concept, most similar and dissimilar instances to a given instance $x_a \in D_t$ are defined as follows:

Definition 6.4 Let $x_a \in X_t \times Y \subseteq D_t$, Sm and Dm are $(\varepsilon, \gamma, B, \pi)$- good similarity and dissimilarity functions, $U_D(\cdot)$ is the classifier constructed using the instances of a given domain D. $\beta, \partial > 1/2$, the sets of most similar $KS\_\beta_D(x_a)$ and most dissimilar $KD\_\partial_D(x_a)$ instances in a given domain D to x_a are defined as follows:

$KS_{\beta_D}(x_a)$

$= \{x_i \in D |\, Sm(x_a, x_i) \geq \beta, Sm(x_a, x_{i+1}) \geq Sm(x_a, x_i), |KS\_\beta_D(x_a)| = K_\beta\},$

$KD_{\partial_D}(x_a)$

$= \{x_i \in D |\, Dm(x_a, x_i) \geq \partial, Dm(x_a, x_{i+1}) \geq Dm(x_a, x_i), |KD\_\partial_D(x_a)| = K_\partial\},$

where

$\forall x_i \in KS_{\beta_D}(x_a), U_D^{\tilde{y}_a}(x_i) > \emptyset \geq 1 - (\varepsilon + \delta)$ and

$\forall x_j \in KD\_\partial_D(x_a), U_D^{-\tilde{y}_a}\left(x_j\right) > \emptyset \geq 1 - (\varepsilon + \delta)$ and

we assume that $\forall x_0 \in D, U_D^{\tilde{y}_a}(x_0) + U_D^{-\tilde{y}_a}(x_0) \leq 1.$

This suggests that the instances with a high value of similarity and dissimilarity to the underlying instance have high membership value $(1-(\varepsilon+\delta))$ in the same and opposite label respectively using the constructed classifier $U_D(\cdot)$.

Given $\tilde{F}_s = \{\tilde{f}_{s_1}, \dots, \tilde{f}_{s_{l_s}}\}$ and $\tilde{F}_t = \{\tilde{f}_{t_1}, \dots, \tilde{f}_{t_{l_t}}\}$ are the fuzzy feature sets for source domain D_s and target domain D_t respectively, where $\tilde{f}_k$ is a fuzzy trapezoidal-shaped membership function for each feature. It is assumed that the feature spaces are the same in both domains in setting one ($\tilde{F} = \tilde{F}_s = \tilde{F}_t$). DIC, which is a novel self-organizing clustering technique, is applied to create the trapezoidal-shape fuzzy features. DIC is a dynamic clustering technique that avoids drawbacks such as *stability-plasticity* and *inflexibility* found in other methods and computing trapezoidal-shaped fuzzy sets (Tung et al. 2004). Given $X_s = \left\{x_{s_1}, \dots, x_{s_{n_s}}\right\}$ are the labeled source domain instances and, $X_t = \left\{x_{t_1}, \dots, x_{t_{n_t}}\right\}$ are the unlabeled target domain instances. Given $\tilde{Y} = \{\tilde{y}_1, \tilde{y}_2\}$ is the predictive fuzzy label set of negative and positive classes, which is the same for both domains. Given $f_D(.)$ is a prediction model which is trained by instances from a given domain D, so that $f_D(x_i) = \left\{f_D^{\tilde{y}_1}(x_i), f_D^{\tilde{y}_2}(x_i)\right\} = \left\{\mu_{\tilde{y}_1}(y_a), \mu_{\tilde{y}_2}(y_a)\right\}$ is the vector of membership values of x_i in each class label. Given $Sm(.)$ and $Dm(.)$ are $(\varepsilon, \gamma, B, \pi)$- good similar and dissimilar functions respectively defined in Definition 3.1.2. The MSFBR algorithms in the Fuzzy Bridged Refinement Domain Adaptation (FBRDA) approach are described in following sections.

6.3.1.1 Multi-Step Fuzzy Bridge Refinement Algorithm Type I

We introduce the refinement function Type I which is computed by applying the similarity/dissimilarity based classifier proposed in Definition 6.3. This measure is used in the Step 4-3 of the proposed MSFBR algorithm Type I to refine the labels.

Definition 6.5 Let $z_a \in X_t \times \tilde{Y}$, where $x \in X$ is in fuzzy sets $\tilde{f}_k,\ k = 1, \dots, l$, $f_{D_s}(x_a)$ is the initial label value for instance z_a which is computed by a prediction model

(·) trained by instances from source domain , $U_D(x_a)$ is the label value for instance z_a which is computed by classifier U constructed by instances from a given domain D. The refinement function Type I, is defined as follows:

1 ,

where $RMVI_D(x_a) = \{RMVI_D{}^1(x_a), RMVI_D{}^2(x_a)\}$ are the membership label values of $x_a$and α is a tradeoff coefficient.

<*Multi-Step Fuzzy Bridge Refinement Algorithm Type I*>

Input: *Source domain: D*

Target domain: D

Fuzzy feature space: $\tilde{F}$

Predictive fuzzy labels: $\tilde{Y}$

Prediction model: $f($

Similarity/dissimilarity based classifier:

Coefficient parameter: α

Number of steps: m

Output: *Label (x*

[Begin]

Step 1: Fuzzify the crisp-value of instances from both domains using the Singleton fuzzifier as follows:

where $\tilde{x}$ is the fuzzified equivalent of crisp input x.

Step 2: Perform antecedent matching of fuzzyfied inputs x_i against fuzzy trapezoidal-shaped features $\widetilde{F}_j$, the input membership value in each feature is computed as follows:

$$\mu_{\tilde{F}_j}(\tilde{x}_i) = \begin{cases} 0, & \text{if} \quad x_i \le l_{ij} \\ \frac{x_i - l_{ij}}{u_{ij} - l_{ij}}, & \text{if} \quad l_{ij} \le x_i \le u_{ij} \\ 1, & \text{if} \quad u_{ij} \le x_i \le v_{ij} \\ \frac{r_{ij} - x_i}{r_{ij} - u_{ij}}, & \text{if} \quad v_{ij} \le x_i \le r_{ij} \\ 0, & \text{if} \quad x_i \ge r_{ij} \end{cases} \quad i \in \{1,2,\dots,n\}, \quad j \in \{1,2,\dots,l\}.$$

Step 3: Train the prediction model by the labeled instances of source domain $f_{D_s}(\cdot)$.

Step 4: Compute the matrix of refined membership value of target domain instances in class labels $RMVI_{n_t \times 2}$ using the similarity/dissimilarity classifier constructed by mixture domain D_w.

For $w = 1$ to m

Step 4-1: Create a mixture domain of source and target domain in each step.

$$D_w = \left(1 - \frac{w}{m}\right)|D_s| + \frac{w}{m}(|D_t|)$$

Step 4-2: Construct $U_{D_w}(\cdot)$ by the positive and negative instances from domain D_w.

Step 4-3: Refine the membership value of target instances in class labels using refinement function Type I.

For $i = 1$ to n_t

For $j = 1$ to 2

$$MRVI_{D_w}(i,j) = \alpha U_{D_w}^{\tilde{y}_j}(x_{t_i}) + (1-\alpha) U_{D_w}^{\tilde{y}_j}(x_{t_i})$$

Next j

Next i

Next w

[End]

The defined refinement function (Definition 3.2) is applied in the proposed algorithm through a multi-step path. The given instance comes from the target domain while the classifier is trained by positive and negative instances which come from a set of mixture domains composed of target and source domains in each step. The first mixture domain is composed of labeled instances in the source domain and unlabeled

instances in the target domain. Through the steps, the number of source instances reduces and the number of target instances increases. The classifier computes the label value of the given target instance based on the positive and negative instances of mixture domains which transform from the source domain to the target domain and bridge the gap between the two domains.

6.3.1.2 MULTI-STEP FUZZY BRIDGE REFINEMENT ALGORITHM TYPE II

We introduce the refinement function Type II which is computed by applying only the most similar instances to the target instance. This measure is used in the Step 4-4 of the proposed MSFBR algorithm Type II to refine the labels.

Definition 3.1.2.1 Let $z_a = (x_a, y_a) \in X \times \tilde{Y} \subseteq D_t$, $z_{b_i} = (x_{b_i}, y_{b_i}) \in KS\_\beta_D(x_a) \subseteq X \times \tilde{Y} \subseteq D$ where $x \in X$ is in fuzzy sets $\tilde{f}_k, k = 1, \dots, l$ and D is a given domain, $f_D(x_a)$ is the initial label value for instance z_a which is computed by a prediction model $f_D(\cdot)$ trained by instances from D, Sm is $(\varepsilon, \gamma, B, \pi)$ −good similarity function and $U_D(\cdot)$ is the classifier constructed using instances from D. The refinement function Type II, $\mathrm{RMVII_D}(\cdot)$, is defined as follows:

$$RMVII_D(x_a) = \alpha\left(\frac{\sum_{i=1}^{K_\beta} Sm\left(x_a, x_{b_i}\right)\left(U_D\left(x_{b_i}\right) - f_D(x_a)\right)}{K_\beta} + f_D(x_a)\right) + (1-\alpha) f_{D_s}(x_a),$$

where $RMVII_D(x_a) = \{RMVII_D^{\tilde{y}_1}(x_a), RMVII_D^{\tilde{y}_2}(x_a)\}$ are refined membership values of x_a in classes $\tilde{y}_1$and $\tilde{y}_2$ respectively and $0 < \alpha < 1$ is the tradeoff coefficient.

<Multi-Step Fuzzy Bridged Refinement algorithm Type II>

Input: *Source domain:* D_s
Target domain: D_t
Fuzzy feature space: $\tilde{F}$
Predictive fuzzy labels: $\tilde{Y}$
Prediction model $f(.)$
Similarity based classifier: $U(\cdot)$
Coefficient parameter: α
Number of steps: m

Output: Label $(x_{t_i}) = arg\,max_j \{RMVII_{D_t}(i,j) | j = 1,2\}$
[Begin]

Step 1: Fuzzify the crisp-valued of instances from both domains using the Singleton fuzzifier.

$$\mu_{\tilde{X}_s}(\tilde{x}_{s_i}) = \begin{cases} 1, & if\ \tilde{x}_{s_i} = x_{s_i} \\ 0, & Otherwise \end{cases} \quad i \in \{1,2,\dots,n_s\},$$

$$\mu_{\tilde{X}_t}(\tilde{x}_{t_i}) = \begin{cases} 1, & if\ \tilde{x}_{s_i} = x_{s_i} \\ 0, & Otherwise \end{cases} \quad i \in \{1,2,\dots,n_t\},$$

where $\tilde{x}$ is the fuzzified equivalent of crisp input x.

Step 2: Perform antecedent matching of fuzzyfied inputs x_i against fuzzy trapezoidal-shaped features $\widetilde{F}_j$, the input membership value in each feature is computed as follows:

$$\mu_{\tilde{F}_j}(\tilde{x}_i) = \begin{cases} 0, & \text{if} \quad x_i \le l_{ij} \\ \frac{x_i - l_{ij}}{u_{ij} - l_{ij}}, & \text{if} \quad l_{ij} \le x_i \le u_{ij} \\ 1, & \text{if} \quad u_{ij} \le x_i \le v_{ij}\ , \quad i \in \{1,2,\dots,n\},\ \ j \in \{1,2,\dots,l\}. \\ \frac{r_{ij} - x_i}{r_{ij} - u_{ij}}, & \text{if} \quad v_{ij} \le x_i \le r_{ij} \\ 0, & \text{if} \quad x_i \ge r_{ij} \end{cases}$$

Step 3: Train the prediction model by the labeled instances of source domain $f_{D_s}(\cdot)$.

Step 4: Compute the matrix of refined membership value of target domain instances in class labels $RMVII_{n_t \times 2}$ using most similar instances in mixture domain D_w.

For $w = 1\ to\ m$

Step 4-1: Create a mixture domain of source and target domain in each step

$$D_w = \left(1 - \frac{w}{m}\right)|D_s| + \frac{w}{m}(|D_t|)$$

Step 4-2: Construct classifier $U_{D_w}(\cdot)$ using the positive and negative instances from domain D_w

Step 4-3: Compute the set of most similar $KS_{\beta_{D_w}}(x_{t_i})$ instances in domain D_w to each instance in target domain for given $\beta > 0$.

For $i = 1$ to n_t

$$KS\_\beta_{D_w}(x_{t_i}) = \{n_1^i, \dots, n_{k_\beta}^i\}$$

Next i

Step 4-4: Refine **t**he initial membership value of each target instance in class labels using refinement function Type II.

Do

For $i = 1$ to n_t

For $j = 1$ to 2

$$RMVII_{\mathrm{D_w}}(i,j)_k = \alpha_{k-1}\left(\frac{\left(\sum_{p=1}^{k_\beta} Sm\left(x_{t_i}, n_p^i\right)\left[U_{\mathrm{D_w}}(n_p^i) - RMVII_{\mathrm{D_w}}(i,j)_{k-1}\right]\right)}{k_\beta} + RMVII_{\mathrm{D_w}}(i,j)_{k-1}\right) + (1-\alpha_{k-1}) f_{D_s}(x_a)$$

Next j

Next i

$K = k + 1$

Until $\boldsymbol{RMVII_{\mathrm{D_w}}}(\cdot)$ converges

Next *w*

[End]

6.3.1.3 Multi-Step Fuzzy Bridge Refinement Algorithm Type III

We introduce the refinement function Type III which is computed by applying the most similar and dissimilar instances to the target instance. This measure is used in the Step 4-3 of the proposed MSFBR algorithm Type III to refine the labels.

Definition 3.1.3.1 Let $z_a = (x_a, y_a) \in X \times \tilde{Y} \subseteq D_t$, $z_{b_i} = (x_{b_i}, y_{b_i}) \in KS\_\beta_D(x_a) \subseteq X \times \tilde{Y} \subseteq D$, $z_{c_i} = (x_{c_i}, y_{c_i}) \in KD\_\partial_D(x_a) \subseteq X \times \tilde{Y} \subseteq D$ where $x \in X$ is in fuzzy sets $\tilde{f}_k, k = 1, \ldots, l$ and D is a given domain, $f_D(x_a)$ is the initial label value for instance z_a which is computed by a prediction model $f_D(\cdot)$ trained by instances from D, Sm and Dm are $(\varepsilon, \gamma, B, \pi)$ −good similarity and dissimilarity functions and $U_D(\cdot)$ is the classifier constructed using instances from D. The refinement function Type III, $RMVIII_D(\cdot)$, is defined as follows:

$$RMVIII_D(x_a)$$

$$= \alpha\left(\frac{\sum_{i=1}^{K_\beta} Sm\left(x_a, x_{b_i}\right)\left(U_D(x_a) - f_D(x_a)\right)}{K_\beta} - \frac{\sum_{i=1}^{K_\partial} Dm\left(x_a, x_{c_i}\right)\left(U_D\left(x_{c_i}\right) - f_D(x_a)\right)}{K_\partial} + f_D(x_a)\right)$$

$$+(1-\alpha) f_{D_s}(x_a),$$

where $RMVIII_D(x_a) = \{RMVIII_D^{\tilde{y}_1}(x_a), RMVIII_D^{\tilde{y}_1}(x_a)\}$ are refined membership values of x_a in classes $\tilde{y}_1$ and $\tilde{y}_2$ respectively and $0 < \alpha < 1$ is the tradeoff coefficient.

<*Multi-Step Fuzzy Bridged Refinement Algorithm Type III*>

Input: *Source domain:* D_s

Target domain: D_t

Fuzzy feature space: $\tilde{F}$

Predictive fuzzy labels: $\tilde{Y}$

Prediction model $f(\cdot)$

Similarity based classifier: $U(\cdot)$

Coefficient parameter: α

Number of steps: m

Output: *Label* $(x_{t_i}) = arg\, max_j \{RMVIII_{D_t}(i,j) | j = 1,2\}$

[Begin]

Step 1: Fuzzify the crisp-valued of instances from both domains using the Singleton fuzzifier.

$$\mu_{\tilde{X}_s}(\tilde{x}_{s_i}) = \begin{cases} 1, & if\ \tilde{x}_{s_i} = x_{s_i} \\ 0, & Otherwise \end{cases} \quad i \in \{1,2,\dots,n_s\},$$

$$\mu_{\tilde{X}_t}(\tilde{x}_{t_i}) = \begin{cases} 1, & if\ \tilde{x}_{s_i} = x_{s_i} \\ 0, & Otherwise \end{cases} \quad i \in \{1,2,\dots,n_t\},$$

where $\tilde{x}$ is the fuzzified equivalent of crisp input x.

Step 2: Perform antecedent matching of fuzzyfied inputs x_i against fuzzy trapezoidal-shaped features $\widetilde{F}_j$, the input membership value in each feature is computed as follows:

$$\mu_{\tilde{F}_j}(\tilde{x}_i) = \begin{cases} 0, & \text{if} \quad x_i \le l_{ij} \\ \frac{x_i - l_{ij}}{u_{ij} - l_{ij}}, & \text{if} \quad l_{ij} \le x_i \le u_{ij} \\ 1, & \text{if} \quad u_{ij} \le x_i \le v_{ij}\ ,\ i \in \{1,2,\dots,n\}, \quad j \in \{1,2,\dots,l\} \cdot \\ \frac{r_{ij} - x_i}{r_{ij} - u_{ij}}, & \text{if} \quad v_{ij} \le x_i \le r_{ij} \\ 0, & \text{if} \quad x_i \ge r_{ij} \end{cases}$$

Step 3: Train the prediction model by the labeled instances of source domain $f_{D_s}(\cdot)$.

Step 4: Compute the matrix of refined membership value of target domain instances in class labels $RMVIII_{n_t \times 2}$ using most similar instances in mixture domain D_w.

For $w = 1\ to\ m$

Step 4-1: Create a mixture domain of source and target domain in each step

$$D_w = \left(1 - \frac{w}{m}\right)|D_s| + \frac{w}{m}(|D_t|)$$

Step 4-2: Construct classifier $U_{D_w}(.)$ using the positive and negative instances from domain D_w

Step 4-3: Compute the sets of most similar and dissimilar instances, $KS\_\beta_{\mathrm{D_w}}(x_{t_i})$ and $KD\_\partial_{\mathrm{D_w}}(x_{t_i})$, in domain D_w to each instance in target domain for given $\beta, \partial > 0$.

For $i = 1$ to n_t

$KS\_\beta_{D_w}(x_{t_i}) = \{n_1^i, \dots, n_{k_\beta}^i\}$

$KD\_\partial_{D_w}(x_{t_i}) = \{v_1^i, \dots, v_{k_\partial}^i\}$

Next i

Step 4-4: Refine **t**he initial membership value of each target instance in class labels using refinement function Type III.

Do

For $i = 1$ to n_t

For $j = 1$ to 2

$$RMVIII_{\mathrm{D_w}}(i,j)_k$$
$$= \alpha_{\mathrm{k-1}}\left(\frac{\left(\sum_{p=1}^{k_\beta} Sm\left(x_{t_i}, n_p^i\right)\left[U_{\mathrm{D_w}}\left(n_p^i\right) - RMVIII_{\mathrm{D_w}}(i,j)_{k-1}\right]\right.}{k_\beta}\right.$$
$$\left. - \frac{\left(\sum_{p=1}^{k_\partial} Dm\left(x_{t_i}, v_p^i\right)\left[U_{\mathrm{D_w}}\left(v_p^i\right) - RMVIII_{\mathrm{D_w}}(i,j)_{k-1}\right]\right.}{k_\partial}\right.$$
$$\left. + RMVII_{\mathrm{D_w}}(i,j)_{k-1}\right)$$
$$+(1-\alpha_{k-1}) f_{D_s}(x_a)$$

Next j

Next i

$K = k + 1$

Until $RMVIII_{\mathrm{D_w}}(\cdot)$ converges

Next w

[End]

As can be seen, the refinement functions Type II and III are based on the fact that the label of the most similar and dissimilar instances to the target instance can be used to modify the initial label of target instance, which was initialized by a prediction model trained by source domain instances. The labels of the most similar and dissimilar instances are achieved by the classifier, which is constructed on the positive and negative instances of mixture domain of source and target domains. As the result of the MSFBR algorithms, a label matrix for all unlabeled instances of the target domain is achieved. Each row of this matrix indicates the membership values of one instance in all label classes. The MSFBRs algorithms can be performed at least in the two-step refinement process by specifying $w = 2$, which firstly refines the labels towards $D_w = \frac{1}{2}|D_s| + \frac{1}{2}|D_t|$, and then toward $D_w = D_t$. The results of the two-step FBR algorithms (2SFBR) have demonstrated significant improvement in comparison with initial labels. However, the accuracy of each data set follows the performance of the prediction model and, consequently, has poor performance in some cases, which will be described in the Experiments section. To solve these problems and improve the

predictive accuracy, we propose having multiple steps to refine the initial labels. The refinement process moves from the source domain (D_s) toward target domains (D_t) through m steps, which indicates the percentage of instances of the source domain and target domain in the mixture domain in each step of refinement. As w increases, the contribution of source domain data in the mixture domain becomes less and conversely, the portion of target domain data increases. Accordingly, the consequent neighboring mixture domains are similar to each other and smoothly transfer from the source domain toward the target domain. Through the multi-step process, it is possible to make a bridge and transfer the label structure between the source and target domains more accurately and easily.

6.3.2 Feature Alignment-based Cross Domain Adaptation (Second Approach)

In this approach, three feature spaces are first defined: (1) Domain-Independent Feature Space ($DIFS$); (2) Source Domain-Specific Feature Space ($DSFS_s$) and (3) Target-Domain Specific Feature Space ($DSFS_t$). Based on these three feature spaces, the Feature Alignment-based Cross Domain Adaptation (FACDA) approach is conducted in five main phases : (1) A prediction model is trained using labeled source instances based on source domain features and the initial labels of target instances are predicted by this model based on a $DIFS$; (2) The labels of target instances are refined using a MSFBR algorithm; (3) A Fuzzy Genetic Feature Weighting algorithm (Ramze Rezaee et al. 1999; Rhee & Lee 1999) is applied to weight the features in $DSFS_t$ using refined labels; (4) The FSFA algorithm is applied to cluster the features and then weight each feature in $DSFS_t$ based on their correlation; and (5) The significant features in $DSFS_t$ are selected according to the gained weights and the prediction model is retrained using the refined labels on significant $DSFS_t$.

Given $\tilde{F}_s = \{\tilde{f}_{s_1}, \dots, \tilde{f}_{s_{l_s}}\}$ and $\tilde{F}_t = \{\tilde{f}_{t_1}, \dots, \tilde{f}_{t_{l_t}}\}$ are the fuzzy feature sets for source domain D_s and target domain D_t respectively, where $\tilde{f}_k$ is a fuzzy trapezoidal-shaped membership function for each feature. It is assumed that the feature spaces are

different in both domains in setting two ($\tilde{F}_s \neq \tilde{F}_t$). DIC, which is a novel self organizing clustering technique, is applied to create the trapezoidal-shape fuzzy features. DIC is a dynamic clustering technique that avoids drawbacks such as stability-plasticity and inflexibility found in other methods and computing trapezoidal-shaped fuzzy sets (Tung et al. 2004). Given $X_s = \left\{x_{s_1}, \dots, x_{s_{n_s}}\right\}$ are the labeled source domain instances and, $X_t = \left\{x_{t_1}, \dots, x_{t_{n_t}}\right\}$ are the unlabeled target domain instances. Given $\tilde{Y} = \{\tilde{y}_1, \tilde{y}_2\}$ is the predictive fuzzy label set of negative and positive classes, which is the same for both domains. Given $f_D(\cdot)$ is a prediction model which is trained by instances from a given domain D, so that $f_D(x_i) = \left\{f_D^{\tilde{y}_1}(x_i), f_D^{\tilde{y}_2}(x_i)\right\} = \left\{\mu_{\tilde{y}_1}(y_a), \mu_{\tilde{y}_2}(y_a)\right\}$ is the vector of membership values of x_i in each class label. Given Unified Feature Space: $UFS = \{\widetilde{G}_\iota | \widetilde{G}_\iota \in \tilde{F}_s \cup \tilde{F}_t\}$ is the union of features in both domains, Domain-Independent Feature Space: $DIFS = \{\widetilde{H}_i | \widetilde{H}_i \in \tilde{F}_s \cap \tilde{F}_t\}$ is the intersection of features in both domains, Domain-Specific Feature Space: $DSFS = \{\widetilde{\Psi}_i | \widetilde{\Psi}_i \in UFS - DIFS\}$, Source Domain-Specific Feature Space: $DSFS_s = \left\{\widetilde{\Phi}_i \middle| \widetilde{\Phi}_i \in \tilde{F}_s - DIFS\right\}$, Target Domain-Specific Feature Space: $DSFS_t = \{\widetilde{\Omega}_i | \widetilde{\Omega}_i \in \tilde{F}_t - DIFS\}$. The Feature Alignment-based Cross Domain Adaptation (FACDA) approach is described as follows.

6.3.2.1 PHASE ONE

A prediction model such as NN, SVM or FNN (Behbood et al. 2010) is chosen as a prediction model $f_D(\cdot)$. It is trained on source domain features ($\tilde{F}_s$) using labelled instances data in the source domain $f_{D_s}(\cdot)$. The number of inputs of the model is equal to the number of features of the source domain and it has one output to assign a binary label to the instance.

6.3.2.2 PHASE TWO

The prediction model which is trained in the previous phase is applied to predict the labels of the target instances using the data which is projected on $DIFS$. This means that the number of inputs of the prediction model is the number of features in $DIFS$

and the value of the target instances in these features is applied for prediction. Although the features in $DIFS$ are similar in both domains, they have different distribution, so a MSFBR algorithm is applied to refine the target predicted labels and achieve better predictive accuracy.

6.3.2.3 PHASE THREE

The FSFA algorithm, which is based on graph spectral theory, is applied. Graph spectral theory assumes that if two nodes in a graph are connected to many common nodes, they should be similar. In the proposed algorithm, we assume that if two fuzzy feature sets of $DSFS$ are highly connected to many common fuzzy feature sets of $DIFS$, they are most probably related to each other and will be aligned with the same cluster with a specific membership value. The proposed algorithm returns a set of membership values for each fuzzy feature set in each cluster. Based on the membership value, we can assign a primary weight (ω_1) to the most meaningful and significant features of $DSFS$ to reduce the gap between two domains and improve prediction accuracy. Before applying the algorithm, we need to find a number of initial parameters. First, we should identify the number of clusters N_c and their cores. Since we aim to cluster features of $DSFS$ based on their relation to features of $DIFS$, the number of clusters is equal to the number of features of $DIFS$ ($N_c = |V_{DIFS}|$) and each feature of $DIFS$ is considered as a core for each cluster.

< Fuzzy Spectral Feature Alignment Algorithm>

Input: *Source domain:* D_s

Target domain: D_t

Target Fuzzy feature space: $\tilde{F}_t$

Source Fuzzy feature space: $\tilde{F}_s$

Predictive fuzzy labels: $\tilde{Y}$

Domain-Independent Feature Space: $DIFS$

Domain-Specific Feature Space: $DSFS$

Source Domain-Specific Feature Space: $DSFS_s$

Target Domain-Specific Feature Space: $DSFS_t$

Prediction model $f(\cdot)$

Output: $\omega_1(\Psi_{ij}) = U_c(\Psi_{ij})$ *is the membership value of each fuzzy linguistic term of DSFS in each* cluster.

[Begin]

Step 1: Build Fuzzy Correlation Coefficient matrix

$$FCC_{|V_{DSFS}|\times|V_{DIFS}|} = [FCC_{\Psi_{ij},h_{ij}} | FCC_{\Psi_{ij},h_{ij}} = \frac{\sum_{k=1}^{n}\left(\mu_{\Psi_{ij}}(x_k)-\bar{\mu}_{\Psi_{ij}}\right)\left(\mu_{h_{ij}}(x_k)-\bar{\mu}_{h_{ij}}\right)}{(n-1)\left(S_{\Psi_{ij}}S_{h_{ij}}\right)}] \cdot$$

Step 2: Build FSFA Weight matrix (W)

$$W_{(|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|)} = \begin{bmatrix} |V_{DSFS}| & |V_{DIFS}| \\ \widetilde{0} & \widetilde{FCC} \\ FCC^T & 0 \end{bmatrix}_{(|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|)}$$

Step 3: Build Diagonal matrix (D)

$$D_{(|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|)} =$$

$$\left[D_{i,j} = \begin{cases} \sum_{s=1}^{|V_{DSFS}|+|V_{DIFS}|} w_{p,s} \quad p = 1, \dots, |V_{DSFS}| + |V_{DIFS}|, & i = j \\ 0, & i \neq j \end{cases} \right].$$

Step 4: Establish Transition matrix (P)

$$P = I + [W - D]\frac{\gamma_1}{\max_{i=1,\dots,|V_{DSFS}|+|V_{DIFS}|} D_{i,i}} \cdot$$

where $I \in \mathbb{R}^{|V_{DSFS}|+|V_{DIFS}|)\times(|V_{DSFS}|+|V_{DIFS}|}$ is the identity matrix and $\gamma_1 \in (0,1)$, which is an internal parameter, ensures that all entries of transition matrix are nonnegative. Its default value is 0.1.

Step 5: Establish Alternative matrixes

Let Ψ_o be a fuzzy linguistic term of $DSFS_t$. To assign its membership value ($U_c(\Psi_o)$) to cluster $c_k, k = 1, \dots, N_c$, an alternative weight matrix $\overline{W}$ is formed using original weight matrix W with the row and column of Ψ_o replaced by row and column of h_o which is the core of cluster c_k. Using $\overline{W}$, matrixes $\bar{D}$ and $\bar{P}$ are computed by Equations in Steps 3and 4 respectively.

Step 6: Calculate Diffusion Distance (DD)

$\alpha = \left\lfloor \frac{\gamma_2}{|log\beta|} \right\rfloor$,

where β is the second largest eigenvalue of P, $\lfloor \cdot \rfloor$ denotes the integer part and $\gamma_2 \in (0, \infty)$ is an internal parameter and its default value is 1.

$$DD(\Psi_o, c_k) = \begin{cases} \left|\left|P^\alpha e - \bar{P}^\alpha e\right|\right| \text{where} \quad e(j) = 1, \quad \text{if} \quad j = index(\Psi_o) \\ \left|\left|P^\alpha e - \bar{P}^\alpha e\right|\right| \text{where} \quad e(j) = 0, \quad \text{if} \quad O.W \end{cases},$$

where $\left|\left|. \right|\right|$ is the Euclidean norm.

Step 7: Calculate Membership value

$$U_c(\Psi_o) = \left\{ u_{c_k}(\Psi_o) \middle| u_{c_k}(\Psi_o) = \frac{DD(\Psi_o, c_k)}{\sum_{l=1}^{N_c} DD(\Psi_o, c_l)}, k = 1, \dots, N_c \right\}.$$

Repeat Steps 5 to 7 for each $\Psi_{ij} \in DSFS$.

[End]

6.3.2.4 PHASE FOUR

Since some features are highly correlated and/or irrelevant to the objective task, weighting and selecting the significant features is desirable. In our case, detecting significant features in the target domain by using the labels predicted by similar features in the source domain (objective task) helps us to find the important features based on the similarity between domains, and to transfer the knowledge from the source domain to the target domain. In a typical fuzzy classification or prediction model, each feature is represented by a number of fuzzy linguistic terms like LARGE, MEDIUM and SMALL, and the model is explicitly explained by a number of fuzzy if-then rules, such as: If X is SMALL then Y is MEDIUM where X and Y are features. Hence, weighting the fuzzy linguistic terms of each feature based on their importance in prediction forms the optimum prediction model. In the method proposed by Rezaee et al. (1999), which is applied in this approach, an optimal subset of fuzzy linguistic terms is selected by using conventional search techniques. Instead of a conventional research technique, we use a faster and more efficient weighting method based on a fuzzy-genetic approach, as proposed by Rhee and Lee (1999). The Fuzzy Genetic Feature Weighting (FGFW) algorithm applies the instances in the target domain to the labels which were refined by a MSFBR algorithm in Phase Two to assign a

secondary weight (ω_2) to each fuzzy linguistic term of $DSFS_t$. It is described in detail as follows:

< *Fuzzy Genetic Feature Weighting Algorithm*>

Input: *Refined labels of target instances computed in Phase 2:* $Label\ (x_{t_i})$

Population Number: PN

Crossover Probability: CP

Mutation Probability: MP

Bit Length: BL

Selection Threshold: ST

Error Threshold: ET

Output: $\omega_2(F_{ij}) = J(F_{ij})$ *is the weight value of each fuzzy linguistic term in* D_t

[Begin]

Step 1: Initialization

Initialize algorithm parameters: $PN;\ CP;\ MP;\ BL;\ ST$

Step 2: Project the original data to fuzzy space and generate random population

$\mu_{\tilde{F}_t}(X_t) = \{\mu_{\tilde{F}_{t_1}}(X_t), \dots, \mu_{\tilde{F}_{t_{l_t}}}(X_t)\}$

then a random population of *PN* chromosomes is generated. Each chromosome, which is representative of each fuzzy linguistic term, consists of *BL* gens.

For $s = 1$ to PN

Step 3: Calculate the secondary weight and compute the fitness function value

3.1: $J^s(\mathrm{F}_{ij}) = \frac{Decimal\ value\ equivalent\ to\ binary\ string\ of\ each\ chromosome}{2^{BL}-1}$.

3.2: Execute the prediction model with lingustic terms $\{\mathrm{F}_{ij} | J^s(\mathrm{F}_{ij}) > ST\}$ using $(x_{t_i}, Label\ (x_{t_i})), i = 1, \dots, n_t$.

3.3: Calculate fitness function:

$$E(s) = (1 - G(s)) = 1 - \sqrt{\frac{TP(s)}{TP(s)\times FN(s)} \times \frac{TN(s)}{FP(s)\times TN(s)}}\ .$$

Next s

Repeat

Step 4: Reproduction

4.1: Select two parent chromosomes according to their values of $E(i)$. (the lower the value of $E(i)$, the more probability there is that it will be selected)

4.2: Cross over the selected parents using CP to form a new offspring.

4.3: Mutate new offspring at each locus using MP.

4.4: Place new offspring in a new population.

Until $|\text{new population}| < NP$
If $\exists s,\ E(s) < ET$

Then return $\omega_2(\mathrm{F}_{ij}) = J^s(\mathrm{F}_{ij})$

Else Go to Step 3

[End]

6.3.2.5 PHASE FIVE

The primary and secondary weights gained in previous phases are combined to achieve the final weight for each fuzzy linguistic term in the target domain. According to the final weight, the most significant linguistic terms are elected to train the prediction model and make the final prediction. The final weight is computed using β which is the experimental trade-off parameter to find the optimal combination of primary and secondary weights. The value of β is empirically achieved in each experiment to achieve maximum accuracy: $= \beta\omega_1 + (1-\beta)\omega_2$. The final weight (ω) of each fuzzy linguistic term is compared with a predefined threshold (ε) and if it satisfies the threshold ($\omega > \varepsilon$), then the corresponding linguistic terms are selected to participate in training the prediction model. The selected linguistic term is denoted as the Selected Target-Domain Feature Space (STDFS) and is presented as $\Lambda = \{\Lambda_i |\ \omega(\Lambda_i) > \varepsilon\}$. The prediction model is trained on STDFS (Λ) using the labeled target instances data (gained in Phase Two).

6.4 EXPERIMENTS AND ANALYSIS

This section presents a set of experiments to validate the two proposed approaches using real-world bank failure data in which the prediction label has two classes: failed

and survived. A number of experiments to examine the performance of approaches to transfer knowledge from the United States banking system to the Australian banking system are performed. The predictive accuracy of the proposed approaches is examined using different baselines and is benchmarked against other similar existing methods. The results demonstrate a significant improvement which is proved by statistical tests.

6.4.1 DATA SETS

The data sets used in the experiments are divided into two domains: (1) United States bank failure data (source domain); (2) Australian bank failure data (target domain). The source domain data set and financial variables used in the experiments are extracted from Call Report Data, which is downloaded from the website of the Federal Reserve Bank of Chicago[12] and the status of each bank is identified according to the Federal Financial Institutions Examination Council (FFIEC)[13]. The data set includes the observation period of the survived banks of 21 years from Jun 1980 to Dec 2000, based on the history of each bank in FFIEC. There are 548 failed banks and 2555 survived ones. Although Tung et al. (Tung et al. 2004) used nine financial features according to their statistical significance and correlation, it is observed that the model with three features has less created rules and less computational load. Each feature is ranked based on the importance of a feature as a result of a future selection process and three features (identified by a star in the tables) with the highest grade are selected (Ng et al. 2008). The definitions of all features are described in Table 2.2. Another data-set has been obtained from Thomson Reuters DataStream of Australian Security Exchange market[14]. This data set includes historical data of 15 Australian banks over 30 years from Jan 1982 to Dec 2000. There are 6 failed and 9 survived banks which are identified in the database. Thirty two financial ratios are identified for this data set. These features, which measure the CAMEL ranking system, are very

[12] http://www.chicagofed.org
[13] http://www.ffiec.gov/nicpubweb/nicweb/NicHome.aspx
[14] http://online.thomsonreuters.com/datastream/

popular and are applied by many studies in bank failure prediction. The definitions of these features, which also include some of the United States financial features, are provided in Table 2.1. As it can be seen, six financial ratios (highlighted by bold font in tables) are the same in both domains. According to the data sets, there are very few Australian banks existed in the data set. Since applying machine learning prediction methods requires a training process with an adequate historical data-set, utilizing the learning-based method for Australian banks brings poor results. However there are so many American cases available to be used for Australian bank failure prediction. Therefore this situation clearly addresses the problem which this study aims to solve: a bank failure prediction model with the ability to transfer knowledge from cases in the United States (Source Domain) and to be used for Australian cases (Target Domain).

6.4.2 RESEARCH DESIGN AND COMPARISON

The instances, which are selected from the data sets up to year 1998, are considered as training data. The task is the prediction of banks' financial status in years 1998, 1999 and 2000 i.e., 0, 1 and 2 years respectively from 1998. To reduce the influence of the imbalanced data sets problem in the United States data set, the SMOTE (Chawla et al. 2002) is applied to the training data set. The number of failed banks increases to the number of survived banks to achieve a balanced data set, which improves the accuracy of prediction without losing important information. In each experiment, the training data set splits into two pools: (1) failed banks denoted (positive class) with output "+1"; (2) survived banks (negative class) denoted with output "-1". The 5-fold cross validation method is applied for training. The predictors are trained using training data sets and then evaluated by the testing data sets. The accuracy of the experiment in each scenario, which is the mean accuracy of cross-validation groups, is measured and calculated by GM.

To specify the similarity and dissimilarity functions and construct the classifier, we follow the approach introduced in (Wang et al. 2008). The proposed approach, called

DBoost, is applied to find the labels of similar and dissimilar instances using Euclidean distance. DIC is applied to create the fuzzy features. DIC is a dynamic clustering technique which avoids drawbacks such as stability-plasticity and inflexibility found in other methods and computing trapezoidal-shaped fuzzy sets (Tung et al. 2004). We perform both approaches on the labels predicted by the prediction models, from which we receive the unrefined (initial) labels of target instances. To ensure that the proposed approaches are sufficiently robust and are not dependent on the prediction model, we select five different prediction models in the experiments: (1) Naïve Bayes (Caruana & Niculescu-Mizil 2006) which performs remarkably well in much of the research, despite its simplicity; (2) Support Vector Machine (SVM) (Xing et al. 2007) which is a powerful supervised learning algorithm. In the experiments SVM with linear kernel is used and all options set by default; (3) Multi Layer Perception Neural Network (MLP-NN) (Lin & Lee 1996) which is a popular prediction model; (4) TSVM (Joachims 1999b) which is a state-of-the-art semi-supervised learning algorithm; and (5) Fuzzy Neural Network (FNN) (Behbood et al. 2010) proposed by the authors for bank failure prediction.

The experiments can be divided into two main sections. The first section examines the first approach in which t is assumed that the feature space of both domains is the same but that the distribution of data is varied. The second section examines the second approach, which assumes that the feature spaces of both domains are different but that domains are related.

6.4.2.1 Experiment Design for Bridged Refinement-based Domain Adaptation (First Approach)

In these experiments, we assume that the source domain (United States banking system) and the target domain (Australian banking system) have the same features. We carry out the experiments with two different feature spaces. Since Australian and American data sets have six features in common, we first perform the experiments assuming that these six features are the feature space for both domains. Secondly, we also conduct the experiments with three selected features of nine features which are

identified by a star in Tables 2.1 and 2.2. In these experiments, we assume that both domains have the same feature space with three features.

We perform the experiments using different settings of the proposed algorithms to examine the influence of applying three main factors on the predictive accuracy of algorithms: (1) Fuzzy approach; (2) Similarity and dissimilarity simultaneously; and (3) Multiple steps. The specifications of different settings of the proposed algorithms are described in Table 6.1. As can be seen, we apply the non-fuzzy versions of algorithms to study the contribution of fuzzy approach in predictive accuracy. Also, there are other versions of algorithms which carry out two steps of refinement by specifying $m = 2$ in algorithms. Since the Type I algorithm uses a classifier which is constructed by positive and negative samples of only similar instances or dissimilar instances, it uses the labels of only similar or dissimilar instances to modify the labels of target instances. The Type II algorithm also uses the labels of only similar instances in mixture domains to refine the target instances' labels, while, the Type III algorithm utilizes the labels of similar and dissimilar instances simultaneously to refine the labels of target instances. Also, we apply the 2-Step Bridge Refinement (2SBR) (Xing et al. 2007), which is the closest study to this research, as a baseline in comparisons. The 2SBR, which is a non-fuzzy algorithm, refines the initial target instances' labels through 2 steps of refinement by using only similar instances. In all, we conduct 300 experiments to evaluate the approach.

TABLE 6.1: SETTING SPECIFICATIONS OF PROPOSED ALGORITHMS WHICH ARE USED IN COMPARISONS

Algorithm	Fuzzy Approach	Similarity and Dissimilarity	Multiple Steps
2SBR	×	×	×
2SBR Type III	×	✓	×
MSBR Type II	×	×	✓
MSBR Type III	×	✓	✓
2SFBR Type I	✓	×	×
2SFBR Type II	✓	×	×
2SFBR Type III	✓	✓	×
MSFBR Type I	✓	×	✓
MSFBR Type II	✓	×	✓
MSFBR Type III	✓	✓	✓

6.4.2.2 Experiment Design for Feature Alignment-based Cross Domain Adaptation (Second Approach)

In these experiments, we assume that the source domain (United States banking system) and target domain (Australian banking system) have different feature spaces but with few features in common. The feature space of source domain ($\tilde{F}_s$) is a 9-dimensional space which consists of all the features of the United States banks data set in Table 2.2. The target domain has a 32-dimentional feature space ($\tilde{F}_t$) which is composed of all the features in the Australian banks data set shown in Table 2.1. We take six features (identified by bold font in Tables 2.1 and 2.2) as features which appear in both domains ($DIFS$). The MSFBR Type III is applied as a refinement algorithm in Phase Three of the approach in the experiments.

We evaluate the performance of the FACDA approach; it is benchmarked with those of two existing domain adaptation approaches. Since most existing methods assume that the source and target domains are defined by the same features, they cannot be directly applied to these experiments. Few studies investigate the situation in which domains have different feature spaces, which is called heterogeneous domain adaptation. We apply two recent efficient heterogeneous domain adaptation approaches for comparison in this section, as follows: (1) Manifold Alignment using Correspondences (MAC) (Wang & Mahadevan 2009)**:** The key idea of this approach is to project different domains in a latent space, match the corresponding instances and preserve the topology of each input domain. MAC applies labeled and unlabeled data for domain adaptation and assumes that there are a limited number of labeled data in the target domain. Applying manifold alignment to domain adaptation in this approach needs to specify cross-domain correspondence relationships to learn the mapping function, which may be difficult to gain in most domain adaptation applications; and (2) Manifold Alignment using Labels (MAL) (Wang & Mahadevan 2011): This approach, which is an extension of the previous approach, explores how to use label information rather than correspondence to align input domains. The key

idea underlying this approach is that many source and target domains defined by different feature spaces often share the same labels. Accordingly, MAL learns map functions to project the source and target domains to a new latent space, matches the instances of two domains with the same labels and preserves the topology of each input domain. The approaches are benchmarked against a No-Transfer baseline in which **t**he prediction model is trained by the source domain training data and applied to the target domain using features in DIFS. We compare the domain adaptation approaches with No-Transfer to compute the relatively predictive accuracy growth that these approaches bring about. In conclusion, a total of 60 experiments are performed to evaluate the FACDA approach.

6.4.3 Experiment Results Analysis for Fuzzy Bridged Refinement Domain Adaptation (First approach)

In this section, the results gained from the FBRDA approach experiments are reported in Tables 6.2 to 6.6. To ensure that the proposed algorithms of the FBRDA approach make a significant improvement in accuracy, we perform the proposed algorithms on unrefined labels predicted by five different prediction models: BN; SVM; NN; TSVM; and FNN. In all scenarios, the proposed algorithms improve the accuracy. For instance, in MSFBR Type III, the average relative growth is gained by refining the initial labels predicted by BN, SVM, NN, TSVM and FNN, being 21.86%, 21.92%, 21.86%, 22.80% and 23.47% respectively. Furthermore, the proposed algorithms achieve better accuracy on the feature space with six features as well as three features. For instance, MSFBR Type III algorithm roughly achieves 22.73% and 22.03% relative increase in accuracy in three and six dimensional feature spaces respectively. The algorithms of the FBRDA approach also demonstrate augmentation in accuracy for different time gaps. For instance, if MSFBR Type III algorithm is applied for refining initial prediction, the average relative enhancement in predictive accuracy for years 1998, 1999 and 2000 is 22.13%, 22.38% and 22.63% respectively. According to the overall analysis, it is concluded that the proposed

algorithms significantly improve the predictive accuracy regardless of prediction models, dimension of feature space and the time window of prediction.

TABLE 6.2: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY REFINEMENT ALGORITHMS ON NB

Algorithms	3 features			6 features		
	1998	1999	2000	1998	1999	2000
No-Transfer	66.3357	65.3481	64.4916	67.0717	65.8286	64.9005
2SBR	74.4331	73.1891	72.0501	74.3134	73.5973	72.7022
	12.21%	12.00%	11.72%	10.80%	11.80%	12.02%
2SBR Type III	75.7446	75.1316	74.1163	78.0850	76.2034	75.6666
	14.18%	14.97%	14.92%	16.42%	15.76%	16.59%
MSBR Type II	75.1536	74.4092	73.188	76.1381	75.3842	74.5977
	13.29%	13.87%	13.48%	13.52%	14.52%	14.94%
MSBR Type III	76.9698	75.3039	74.9190	77.7015	76.9553	76.1749
	16.03%	15.24%	16.17%	15.85%	16.90%	17.37%
2SFBR Type I	75.6568	74.4244	73.5318	77.3931	76.5482	75.4394
	14.05%	13.89%	14.02%	15.39%	16.28%	16.24%
2SFBR Type II	77.3084	76.0352	75.2689	77.8753	77.34207	76.4469
	16.54%	16.35%	16.71%	16.11%	17.49%	17.79%
2SFBR Type III	77.1575	76.9329	75.7733	78.8670	77.9769	77.4796
	16.31%	17.73%	17.49%	17.59%	18.45%	19.38%
M-SFBR Type I	76.3086	75.1856	74.9786	78.6278	77.1280	76.6283
	15.03%	15.05%	16.26%	17.23%	17.17%	18.07%
M-SFBR Type II	79.3956	78.4905	77.0609	80.4036	79.5343	78.4055
	19.69%	20.11%	19.49%	19.88%	20.82%	20.81%
M-SFBR Type III	80.4300	78.9855	78.1196	81.8284	80.8546	79.8814
	21.25%	20.87%	21.13%	22.00%	22.83%	23.08%

TABLE 6.3: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY REFINEMENT ALGORITHMS ON SVM

Algorithms	3features			6 Features		
	1998	1999	2000	1998	1999	2000
No-Transfer	67.8320	66.1263	65.7509	67.3039	66.4084	65.8676
2SBR	74.3812	73.7265	72.9608	74.1459	73.6226	72.7005
	12.64%	13.12%	13.47%	11.63%	11.94%	11.98%
2SBR Type III	76.8646	75.7916	75.3744	77.5879	76.0140	75.5293
	16.40%	16.28%	17.22%	16.81%	15.58%	16.33%
MSBR Type II	76.3866	75.5894	74.8549	76.4028	75.4401	74.9400
	15.68%	15.97%	16.41%	15.02%	14.70%	15.42%
MSBR Type III	77.3546	76.8532	76.0754	77.2802	76.627	75.9249
	17.14%	17.91%	18.31%	16.34%	16.51%	16.94%
2SFBR Type I	76.4309	76.1963	74.9996	77.2803	76.2388	75.0145
	15.74%	16.91%	16.64%	16.34%	15.92%	15.54%
2SFBR Type II	77.1683	76.4369	76.0580	77.7717	76.7130	76.1433
	16.86%	17.27%	18.28%	17.08%	16.64%	17.28%
2SFBR Type III	78.6753	77.5199	76.3546	78.6573	76.9231	76.3879
	19.14%	18.94%	18.75%	18.42%	16.96%	17.65%
M-SFBR Type I	77.1950	76.2329	75.5512	77.5154	76.3541	75.6313
	16.90%	16.96%	17.50%	16.70%	16.09%	16.49%
M-SFBR Type II	79.1703	78.2447	77.73162	79.8927	78.9126	77.9862
	19.89%	20.05%	20.89%	20.28%	19.98%	20.12%
M-SFBR Type III	80.4953	80.0627	78.9844	80.3975	79.7613	78.9721
	21.90%	22.84%	22.83%	21.04%	21.27%	21.63%

TABLE 6.4: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY REFINEMENT ALGORITHMS ON MLP-NN

Algorithms	3 Features			6 Features		
	1998	1999	2000	1998	1999	2000
No-Transfer	68.5307	67.0498	66.5614	68.6626	67.7269	66.9042
2SBR	76.1670	74.8394	74.0619	76.0958	75.3496	74.0952
	16.07%	14.67%	14.99%	14.28%	14.50%	0.139972
2SBR Type III	77.4439	75.7470	75.1033	76.6377	76.2839	75.4062
	18.01%	16.06%	16.60%	15.10%	15.92%	16.01%
MSBR Type II	76.4862	74.5526	74.1834	76.4746	75.3778	74.3444
	16.55%	14.23%	15.17%	14.85%	14.54%	14.38%
MSBR Type III	77.6049	76.9493	75.7316	77.3356	76.3625	75.851
	18.26%	17.91%	17.58%	16.14%	16.04%	16.70%
2SFBR Type I	76.3398	75.8307	74.8915	76.2626	75.3779	74.7459
	16.33%	16.19%	16.27%	14.53%	14.54%	15.00%
2SFBR Type II	77.7943	76.9403	75.5715	77.807	76.2758	75.8257
	18.55%	17.89%	17.33%	16.85%	15.91%	16.66%
2SFBR Type III	77.8729	76.6011	75.8666	77.7280	77.0400	76.2214
	18.67%	17.37%	17.79%	16.73%	17.07%	17.27%
M-SFBR Type I	77.0711	75.9193	75.0313	77.3959	76.3778	74.9081
	17.44%	16.33%	16.49%	16.24%	16.06%	15.25%
M-SFBR Type II	78.3555	77.6432	77.1275	78.7881	77.7502	76.8476
	19.40%	18.97%	19.75%	18.33%	18.15%	18.23%
M-SFBR Type III	80.8179	79.9110	78.815	80.6276	79.4904	78.8579
	23.15%	22.44%	22.37%	21.09%	20.79%	21.32%

TABLE 6.5: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY REFINEMENT ALGORITHMS ON TSVM

Algorithms	3 Features			6 Features		
	1998	1999	2000	1998	1999	2000
No-Transfer	68.7507	68.2737	67.3358	69.2587	68.0057	67.1496
2SBR	76.3084	75.4138	74.6441	76.5540	75.9754	74.5480
	16.48%	16.38%	16.52%	14.48%	16.04%	0.1484
2SBR Type III	77.2678	76.8564	75.6983	77.8855	76.3819	75.9621
	17.95%	18.60%	18.16%	16.48%	16.66%	17.02%
MSBR Type II	76.0523	75.2865	74.6812	76.3808	75.2597	74.7471
	16.09%	16.18%	16.57%	14.23%	14.95%	15.15%
MSBR Type III	77.7422	76.6087	76.3713	78.1261	76.9416	76.0428
	18.67%	18.22%	19.21%	16.84%	17.52%	17.15%
2SFBR Type I	77.1204	76.0713	75.1827	77.2464	76.1152	75.4631
	17.72%	17.39%	17.36%	15.52%	16.25%	16.25%
2SFBR Type II	78.5133	77.0787	76.6124	78.0453	77.3069	76.6247
	19.85%	18.95%	19.59%	16.71%	18.07%	18.04%
2SFBR Type III	78.2420	77.6048	76.9348	78.5483	77.5523	76.8984
	19.43%	19.76%	20.09%	17.47%	18.45%	18.46%
M-SFBR Type I	78.1955	77.1846	76.4664	77.6695	76.5369	76.0768
	19.36%	19.11%	19.36%	16.15%	16.90%	17.20%
M-SFBR Type II	79.7289	78.6636	77.7960	79.7148	78.7019	77.5709
	21.70%	21.39%	21.44%	19.21%	20.20%	19.50%
M-SFBR Type III	80.8658	79.8040	79.1705	81.2883	80.4959	79.2579
	23.44%	23.15%	23.58%	21.56%	22.94%	22.10%

TABLE 6.6: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY REFINEMENT ALGORITHMS ON FNN

Algorithms	3 Features			6 Features		
	1998	1999	2000	1998	1999	2000
No-Transfer	69.9107	68.7815	67.7463	69.4659	68.8120	67.7762
2SBR	78.3164	76.6950	76.2631	78.3119	76.9053	76.4751
	18.18%	18.02%	18.27%	17.10%	16.48%	0.182602
2SBR Type III	77.9509	76.2512	75.8682	77.6334	76.520	75.8721
	17.63%	17.34%	17.66%	16.09%	15.90%	17.33%
MSBR Type II	76.4395	75.8158	75.2793	76.6442	75.4741	75.0287
	15.35%	16.67%	16.75%	14.61%	14.32%	16.02%
MSBR Type III	77.8616	77.3053	76.6094	78.4867	76.7884	76.2648
	17.50%	18.96%	18.81%	17.36%	16.31%	17.93%
2SFBR Type I	77.5208	76.6781	75.4897	77.1835	76.2307	75.3269
	16.98%	18.00%	17.07%	15.41%	15.46%	16.48%
2SFBR Type II	77.9149	77.1699	76.4000	78.6057	76.9787	76.2549
	17.58%	18.75%	18.48%	17.54%	16.59%	17.92%
2SFBR Type III	78.8646	77.0715	76.6681	78.5113	77.0995	76.8588
	19.01%	18.60%	18.90%	17.40%	16.78%	18.85%
M-SFBR Type I	78.4129	77.9995	77.0133	79.1726	77.6842	76.9046
	18.33%	20.03%	19.44%	18.39%	17.66%	18.92%
M-SFBR Type II	80.4959	79.7150	78.8913	80.5119	79.1317	78.8525
	21.47%	22.67%	22.35%	20.39%	19.86%	21.94%
M-SFBR Type III	81.5355	80.9830	80.2079	82.1185	80.5849	80.1233
	23.04%	24.62%	24.39%	22.79%	22.06%	23.90%

As discussed in the research design section, we are interested in examining the contribution of three main factors in the improvement, namely (1) Fuzzy approach; (2) Similarity and dissimilarity simultaneously; and (3) Multiple steps. The performances of different settings of the proposed algorithms are compared to evaluate each factor. As can be seen from Tables 6.2 to 6.6, the algorithms which apply fuzzy approach outperform other non-fuzzy algorithms, the multiple-step algorithm achieves better performance than 2-step algorithms, and the algorithms which use similar and dissimilar instances simultaneously are more accurate than those that only utilize similar or dissimilar instances for refinement. To confirm the influence of these factors and growth in accuracy, the Holm test (Holm 1979), which is a non-parametric statistical test, is applied to specify whether the improvement is significant or not. The test is performed on thirty scenarios including five prediction models: BN; SVM; NN; TSVM; and FNN, three time periods of perdition: same year; one year before; and two years before, and two feature spaces: three-dimensional and six-dimensional feature spaces. The results of the statistical tests, which are presented in Tables 6.7 to 6.9, reject almost all hypotheses of equality of the accuracy in 0.05 level of confidence.

Table 6.7 demonstrates the results of Holm tests for examining the influence of the fuzzy approach on predictive accuracy. We compare the performance of fuzzy algorithms with that of non-fuzzy algorithms which have the same status regarding two other factors. As can be seen, all hypotheses are rejected with the exception of Test 1. It is concluded that the fuzzy approach significantly enhances predictive accuracy, particularly when the multiple-step algorithms with similarity and dissimilarity functions are applied for refinement. Tests 4 and 5 show that the multiple-step fuzzy algorithms outperform multiple-step non-fuzzy algorithms, regardless of whether or not the similar and dissimilar instances are simultaneously used for refinement. Tests 2 and 3 imply the same conclusion for 2-step algorithms. Test 1 shows that if the classifier constructed by positive and negative instances is employed for 2-step refinement, the influence of the fuzzy approach is not significant.

We benchmark the performance of different settings of Type III algorithm, which uses similar and dissimilar cases for refinement, with other algorithms which use only similar or dissimilar instances to refine the initial labels.

TABLE 6.7: HOLM TESTS (95% OF CONFIDENCE) EXAMINE THE INFLUENCE OF FUZZY APPROACH ON REFINEMENT ALGORITHM PERFORMANCE

	Comparison	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Hypothesis
1	2SBR vs. 2SFBR Type I	2.6469	0.008	0.003	Not Rejected
2	2SBR vs. 2SFBR Type II	4.1260	3.690E-5	0.002	Rejected
3	2SBR Type III vs. 2SFBR Type III	4.9435	7.674E-7	0.001	Rejected
4	MSBR Type II vs. MSFBR Type II	5.1381	2.775E-7	0.001	Rejected
5	MSBR Type III vs. MSFBR Type III	3.9703	7.176E-5	0.002	Rejected

Table 6.7 shows the Holm tests studying these comparisons. Tests 2 to 6, which reject the hypotheses, demonstrate that the Type III algorithm significantly outperforms Type I and II algorithms, regardless of whether 2-step or multiple-step refinement is applied, and whether a fuzzy or non-fuzzy approach is employed. From Test 1, accepting the hypothesis, it is implied that the Type III algorithm is not remarkably different from 2SBR (Xing et al. 2007) when 2–step non-fuzzy refinement is applied to modify the initial labels.

Table 6.8 shows the results of Holm tests that aim to evaluate the contribution of using multiple steps of refinement in accuracy enhancement. We compare 2-step algorithms with multiple-step algorithms which have the same state in other factors. The Holm tests reject all hypotheses with the exception of Test 1. These results demonstrate that using multiple step refinement significantly augments the predictive accuracy particularly when fuzzy approach is applied in refinement. Test 1 accepts the hypothesis, which means that the performances of 2-step and multiple-step non-fuzzy algorithms using similar cases in refinement are not remarkably different.

TABLE 6.8: HOLM TESTS (95% OF CONFIDENCE) EXAMINE THE INFLUENCE OF SIMILARITY AND DISSIMILARITY FUNCTIONS ON REFINEMENT ALGORITHM PERFORMANCE

	Comparison	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Hypothesis
1	2SBR vs. 2SBR Type III	2.4523	0.0142	0.003	Not Rejected
2	MSBR Type II vs. MSBR Type III	4.5542	5.258E-6	0.002	Rejected
3	2SFBR Type I vs. 2SFBR Type III	3.931	8.444E-5	0.002	Rejected
4	2SFBR Type II vs. 2SFBR Type III	5.4106	6.282E-8	0.001	Rejected
5	MSFBR Type I vs. MSFBR Type III	3.5811	3.422E-4	0.002	Rejected
6	MSFBR Type II vs. MSFBR Type III	4.749	2.046E-6	0.001	Rejected

TABLE 6.9: HOLM TESTS (95% OF CONFIDENCE) EXAMINE THE INFLUENCE OF MULTIPLE STEPS ON REFINEMENT ALGORITHM PERFORMANCE

	Comparison	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Hypothesis
1I	2SBR vs. MSBR Type II	0.2725	0.7853	0.05	Not Rejected
2	2SBR Type III vs. MSBR Type III	3.4254	6.1390E-4	0.0022	Rejected
3	2SFBR Type I vs. MSFBR Type I	3.5032	4.5963E-4	0.0021	Rejected
4	2SFBR Type II vs. MSFBR Type II	3.1529	0.0016	0.0023	Rejected
5	2SFBR Type III vs. MSFBR Type III	4.2817	1.8543E-5	0.0017	Rejected

6.4.4 EXPERIMENT RESULTS ANALYSIS FOR FEATURE ALIGNMENT-BASED CROSS DOMAIN ADAPTATION (SECOND APPROACH)

In this section, the FACDA approach is benchmarked against two heterogeneous domain adaptation approaches, namely MAC and MAB. The reason for choosing these approaches is that they achieved the best performance in previous experiments, particularly when the problem included a small number of features, as there are few features in the banking data set. The experiments are carried out according to the experiment design described in Section 6.4.2.2. The results of these experiments are demonstrated in Tables 6.11 to 6.15. The final weights of features, when different final prediction models are applied, are shown in Tables 6.10. Table 6.16 shows the statistical tests which have been conducted to compare the performance of the FACDA approach with that of other approaches.

The proposed FACDA approach calculates the final weight for each feature in the target domain and according to the predefined threshold (ε) in Phase Five, some of the features selected for training. Table 6.10 shows these values when five prediction models are employed. The number of selected features is based on threshold value $\varepsilon = 0.75$. The optimum value for the predefined threshold is achieved experimentally. As can be seen, the final weights of the features and accordingly, the selected features, change when the prediction model varies. For instance, the FACDA approach selects ten and twelve features when SVM and TSVM respectively are prediction models. However, FACDA applies the six features belonging to $DIFS$ for

transfer learning in all scenarios, because they gain high final weights due to their primary weights. Another predefined parameter is the trade-off of parameter β to combine the primary and secondary weight and compute the final weight. The value of is experimentally computed.

TABLE 6.10: FINAL FEATURE WEIGHTS WHEN DIFFERENT PREDICTION MODELS ARE APPLIED

	Prediction Models used in MSFBR Type III Algorithm				
	NB	SVM	NN	TSVM	FNN
Feature 1	**0.78**	**0.81**	**0.79**	**0.79**	**0.80**
Feature 2	0.57	0.57	0.55	0.56	0.55
Feature 3	0.68	0.69	0.68	0.67	0.68
Feature 4	0.58	0.58	0.58	0.57	0.56
Feature 5	0.61	0.60	0.62	0.62	0.60
Feature 6	0.72	0.72	0.71	0.73	0.73
Feature 7	**0.82**	**0.86**	**0.79**	**0.83**	**0.83**
Feature 8	**0.87**	**0.88**	**0.89**	**0.87**	**0.89**
Feature 9	0.68	0.70	0.68	0.68	0.71
Feature 10	**0.89**	**0.89**	**0.88**	**0.89**	**0.91**
Feature 11	0.65	0.65	0.67	0.66	0.65
Feature 12	**0.87**	**0.86**	**0.86**	**0.88**	**0.88**
Feature 13	0.62	0.63	0.61	0.62	0.60
Feature 14	**0.86**	**0.84**	**0.87**	**0.90**	**0.85**
Feature 15	0.70	0.70	0.71	0.72	0.71
Feature 16	0.63	0.63	0.64	0.62	0.64
Feature 17	0.73	0.74	0.72	0.72	0.71
Feature 18	**0.84**	**0.83**	**0.81**	**0.82**	**0.83**
Feature 19	**0.75**	**0.76**	0.74	**0.77**	**0.75**
Feature 20	**0.76**	0.74	**0.75**	**0.76**	**0.75**
Feature 21	0.68	0.69	0.68	0.69	0.69
Feature 22	0.63	0.63	0.63	0.64	0.64
Feature 23	0.56	0.55	0.54	0.56	0.53
Feature 24	0.57	0.57	0.57	0.54	0.56
Feature 25	0.59	0.62	0.61	0.61	0.60
Feature 26	**0.81**	**0.82**	**0.80**	**0.83**	**0.80**
Feature 27	0.66	0.65	0.65	0.66	0.67
Feature 28	0.67	0.65	0.66	0.65	0.65
Feature 29	0.65	0.66	0.66	0.67	0.67
Feature 30	**0.75**	**0.75**	0.74	**0.76**	0.74
Feature 31	0.72	0.73	0.73	0.72	0.71

Feature 32	**0.77**	0.74	**0.76**	**0.75**	**0.76**
N.O. selected features	12	10	10	12	11

Tables 6.11 to 6.15 demonstrate the results of experiments which are carried out to evaluate the performance of the approaches when five prediction models for three periods of predictions are utilized. The approaches are tested for three years of predictions: same year (2000), one year ahead (1999) and two years ahead (1998). According to the results, the accuracy of all the approaches decreases when the period of prediction becomes longer. The proposed FACDA approach significantly enhances the predictive accuracy in all time periods. For instance, the relative increase in accuracy in years 1998, 1999 and 2000 are 16.49%, 17.94% and 18.76% respectively. Moreover, the FACDA improves the accuracy, regardless of which type of prediction model is used. It achieves relative growth in accuracy by roughly 17.77%, 17.78%, 18.04%, 17.66% and 17.40% when BN, SVM, NN, TSVM and FNN respectively are applied. Tables 6.11 to 6.15 show that the proposed FACDA approach outperforms other approaches in all experiments. To examine this improvement more deeply, the Holm test (Holm 1979) is performed using all the accuracy values achieved in experiments for each approach, using a 0.05 level of significance. The results, which are illustrated in Table 6.16, conclude that the proposed approach significantly outperforms other approaches with 95% of confidence. All hypotheses of equality of accuracy are rejected in favor of the FACDA approach.

TABLE 6.11: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY DIFFERENT APPROACHES WHEN BN IS PREDICTION MODEL

Algorithms	1998	1999	2000
No-Transfer	66.335767	65.348189	64.49164
MAC	75.643067	75.347425	74.750685
	14.03%	15.30%	15.91%
MAB	76.271277	75.73823	75.522943
	14.98%	15.90%	17.11%
FACDA	77.302119	77.008348	76.714005
	16.53%	17.84%	18.95%

TABLE 6.12: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY DIFFERENT APPROACHES WHEN SVM IS PREDICTION MODEL

Algorithms	1998	1999	2000
No-Transfer	67.832032	66.126305	65.75092
MAC	77.229264	76.66782	76.271549
	13.85%	15.94%	16.00%
MAB	77.760123	77.498421	76.770177
	14.64%	17.20%	16.76%
FACDA	78.815351	78.313147	78.067504
	16.19%	18.43%	18.73%

TABLE 6.13: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY DIFFERENT APPROACHES WHEN NN IS PREDICTION MODEL

Algorithms	1998	1999	2000
No-Transfer	68.530746	67.049842	66.5614
MAC	77.933936	77.614709	76.940566
	13.72%	15.76%	15.59%
MAB	78.628991	78.502693	77.999807
	14.74%	17.08%	17.18%
FACDA	79.97056	79.56148	79.060913
	16.69%	18.66%	18.78%

TABLE 6.14: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY DIFFERENT APPROACHES WHEN TSVM IS PREDICTION MODEL

Algorithms	1998	1999	2000
No-Transfer	68.750736	68.27374	67.33581
MAC	79.132261	78.628154	78.032699
	15.10%	15.17%	15.89%
MAB	79.330613	79.255191	78.446723
	15.39%	16.08%	16.50%
FACDA	80.48986	80.097705	79.843941
	17.07%	17.32%	18.58%

TABLE 6.15: ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY DIFFERENT APPROACHES WHEN FNN IS PREDICTION MODEL

Algorithms	1998	1999	2000
No-Transfer	69.910732	68.781545	67.74631
MAC	79.287759	78.518045	78.133364
	13.41%	14.16%	15.33%
MAB	79.918691	79.394556	78.807028
	14.32%	15.43%	16.33%
FACDA	81.062068	80.78912	80.468557
	15.95%	17.46%	18.78%

TABLE 6.16: HOLM TESTS (95% OF CONFIDENCE) COMPARISON OF FACDA APPROACH WITH MAC, MAB APPROACHES

	Comparison	$z = (R_0 - R_i)/SE$	p-value	α-Holm	Hypothesis
1	FACDA vs. No-Transfer	6.3639	1.9662E-10	0.008	Rejected
2	FACDA vs. MAC	4.2426	2.2090E-5	0.025	Rejected
3	FACDA vs. MAB	2.1213	0.0339	0.05	Rejected

6.5 SUMMARY

Bank failures cause great negative impact to the financial system and the economy. The early identification of distressed banks and action to implement corrective measures could avoid bankruptcy, which is a critical part of bank risk management. Many machine learning methods have been applied for bank failure prediction; however, one challenge with the machine learning approach is that the training data (source domain) and the test data (target domain) are assumed to have identical feature spaces with underlying distribution. As a result, once the feature space or the feature distribution of the test data changes, the prediction models cannot be used and must be rebuilt and retrained from scratch using newly-collected training data, which is very expensive, if not practically impossible. Similarly, since learning-based machine learning models need adequate labeled data for training, it is nearly impossible to establish a learning-based model for a domain (target domain) which has very few labeled data available for supervised learning.

In the current chapter, we propose domain adaption approaches to transfer and exploit the knowledge from an existing similar but not identical domain (source domain) with plenty of labeled data to construct the prediction model for the target domain. The adopted domain adaptation approaches enable us to take advantage of information in the source domain (US banking system) to predict failures in the target domain (Australian banking system) in which the labeled data is usually very limited. According to the domain adaptation problem definition, two settings for the problem are defined: (1) Domain adaptation and (2) cross-domain adaptation; and two approaches are proposed to solve them. These approaches are the first to use fuzzy technique approach to cope with data uncertainty in financial features. They are independent of the prediction model and they improve the accuracy for any given prediction model. To examine the independence of the proposed approaches, five different prediction models are utilized to compute the initial labels.

The first approach is designed to solve the first problem, in which we assume that the feature spaces of both domains are the same but that the distributions of data differ. This approach takes three critical factors into account in the main proposed algorithm (MSFBR Type III): (1) fuzzy technique approach; (2) Local learning using similar and dissimilar instances simultaneously; and (3) Multi-step refinement in mixture domains. Nine settings composed of three different proposed algorithms are defined to evaluate the contribution of the abovementioned factors in accuracy enhancement. The results of experiments conducted by these settings using data from US and Australian banks suggest that the factors significantly improve predictive accuracy. Based on empirical results, the proposed approach successfully transfers the knowledge from the US banking system to the Australian banking system and predicts the failures with an accuracy of roughly 81%; a 23% relative increase. The second approach aims to solve the second problem in which we assume that the feature spaces of domains are different. It explores the significant features in the target domain by measuring the correlation among features in both domains. It explicitly depicts the relation among source and target domains. The proposed

FACDA approach, which uses MSFBR Type III in phase Two, is benchmarked against two popular existing methods. The statistical tests demonstrate significant superior performance of FACDA in comparison with the performance of other methods. Likewise, FACDA boosts the predictive accuracy remarkably and identifies Australian banks' failure with an accuracy of roughly 80%; a 17% relative increase. It can be concluded that the proposed warning approaches could serve as a useful tool for both banks and financial regulatory authorities.

CHAPTER 7

CONCLUSIONS AND FUTURE STUDY

This chapter draws conclusions on the research presented in this thesis and nominates some future research directions.

7.1 CONCLUSIONS

This research is motivated by the fact that although the FEWS is one of the most interesting fields in business intelligence and attracts many research efforts, the majority of these studies have dealt with financial failure only as a classical prediction model and other crucial features of FEWS have received very limited attention or recognition. Important features of the system such as handling data uncertainty and the class imbalance problem, knowledge generation ability, transferability and flexibility among different but related domains along with inaccurate proposed models, remain open challenges that need to be investigated more extensively and resolved. In light of these issues, this research makes the following main contributions:

(a) It proposes an adaptive inference-based fuzzy neural network (Chapter 3) to achieve research objective 1. The proposed FNN effectively integrates a fuzzy logic-based adaptive inference system with the learning ability of a neural network to generate knowledge in the form of fuzzy rules. It uses a pre-processing phase to deal

with the imbalanced data-sets problem. Additionally, it contains a set of adaptive parametric inference-based learning and rule generation algorithms to handle the imbalanced data-sets problem, reduce prediction error and increase prediction accuracy. A total of 36 experiments have been conducted based on two populations of United States banks to test and validate the proposed algorithm. The results show that the prediction algorithm is very competitively in its accuracy in comparison with three existing financial warning systems: GenSo-EWS (Tung et al. 2004); FCMAC-EWS (Ng et al. 2008); and MLP (Lin & Lee 1996), two popular fuzzy neural networks: ANFIS(Jang 1993); DENFIS (Kasabov & Qun 2002) and one rule learning algorithm: C4.5 (Batista et al. 2004).

(b) It proposes a novel fuzzy domain adaptation method (Chapter 4) called MSFBR to achieve research objectives 1 and 2: dealing with the domain adaptation problem in machine learning. This problem arises when the training data set and the test data set are drawn from different feature distributions. It is the first to utilize fuzzy set techniques to handle vague values of instance features in domain adaptation. Moreover, instead of modifying the baseline model or decision boundary, this study introduces a fuzzy similarity/dissimilarity–based learning method as a local learning for domain adaptation. It explores similar/dissimilar fuzzy instances in the bridged domains and then, using the explored instances, refines the pseudo labels in the test data set that were initially predicted by the shift-unaware prediction model. Sixteen experiments were performed using 20 years of bank failure financial data to evaluate the MSFBR algorithm, and to compare it with existing domain adaptation models. The results demonstrate that the MSFBR method significantly outperforms other models in terms of long-term predictive accuracy. The outputs conclude that the MSFBR algorithm has interesting potential for implementation in financial applications for long-term bank failure prediction.

(c) It proposes a novel fuzzy cross-domain adaptation approach (Chapter 5) to achieve research objectives 4 and 5: solving the cross-domain adaptation problem in which the feature space of source domains differs from that of the target domain. This

approach bridges the gap between source and target domains by aligning domain-specific features with the help of domain-independent features and selecting the significant features in the target domain. It explicitly represents the relationship between the two domains by depicting the correlation between the domain-specific features of both domains through the domain-independent features. Compared with existing models, the proposed approach is more flexible toward the assumptions of probabilistic models and is able to handle the vagueness of feature values. In particular, by modifying the predicted labels in the domain-independent space and co-clustering features in the domain-specific space, the approach solves both distribution difference and feature space difference problems at the same time. The proposed approach is independent of the shift-unaware model and can be applied for different types of prediction models. We have applied this approach to bank failure prediction and the results demonstrate that the proposed fuzzy cross-domain adaptation approach results in a remarkable improvement in predictive accuracy. Its performance is benchmarked against existing popular methods. The results show that it significantly outperforms other methods in different settings.

7.2 FUTURE STUDY

Future directions in this research can be summarized in the following tasks:

(a) As failure prediction incorporates an imbalanced data sets problem, the information granulation concept and techniques can be applied to make a hierarchical fuzzy rule base which can bring about a significant improvement. This is an interesting research direction we will take into account in the future. Also the comparison more recent preprocessing techniques, the application of cost-sensitive learning and the use of measures that takes into account the significance of each class separately will be a future trend of study.

(b) Since many complex problems can be modeled by fuzzy rules which are created by vague data or by expert knowledge, transferring the knowledge in the form of fuzzy rules among such domains is an interesting research direction. These fuzzy

rules are defined based on the fuzzy features and their significance in the discriminative task of the source domains. Therefore, a method which can modify, adjust and transfer the existing rules in the source domains to the target domain is desirable. It will explicitly transfer the knowledge in the form of fuzzy rules instead of feature spaces or distributions. Since it is observed in Chapter 2 that GAs provide a good framework for improving the capabilities of FRBS, they can be used for optimizing and improving FRBS in the rule transfer

(c) Applying the proposed approaches in the case study to other applications and data samples is an attractive research direction. These experiments will justify the generality and accuracy of the approaches and offer remarkable insight to the field of transfer learning.

(d) Previous studies have considered FEWS as a prediction model and it has been referred to as "Business Failure Prediction", "Bank Failure Prediction" and "Bankruptcy Prediction". Decision makers, who are interested in applying FEWS as a Decision Support System in their organizations, expect that FEWS will assist them in decision making processes which aim to prevent financial failure. Hence, the prediction model is only a component of FEWS and a decision making model should be formed as a complementary component to be integrated into FEWS. To form the decision component of the system, a model needs to be developed based on Fuzzy Case-Based Reasoning (FCBR) together with Fuzzy Multi Criteria Decision Making (FMCDM). FCBR, which is an incremental learning method, explicitly processes data to explain analytic results and provides fundamental knowledge for modeling the decision problem of aiding a company that is predicted to fail. Moreover FMCDM, which is a systematic mathematical approach for decision making problems, assists managers and regulators to find optimal solutions and analyze them under given circumstances. In such a situation, a primitive Intelligent Financial Warning Support System framework and model as shown in Figures 7.1 and 7.2, which takes into account both prediction and decision components, can be developed (Behbood & Lu 2011). Fuzzy Neural Network is applied in the prediction component. The input of

this component is the value of financial ratios of the company under consideration. The outputs are the predicted financial status of the company and the fuzzy rule base that explains the reasons for this prediction. These outputs are employed to construct the case base, which is one of the inputs of the decision component. In the decision component, the FCBR and FMCDM are used. The inputs are the case base and expert knowledge which are used to define the decision model and find the solutions. The output of this component takes the form of the suggested solutions based on the prediction results and expert knowledge.

(e) Almost all studies in transfer learning have attempted to solve the domain adaptation problem in classification, clustering, prediction and recognition, yet handling the domain adaptation issue in complex decision making problems remains intact and challenging. To our best knowledge, no study has exploited, transferred and adjusted the knowledge of a given domain to solve the decision making problem in a different but related domain. This is also an appealing research path that warrants exhaustive investigation.

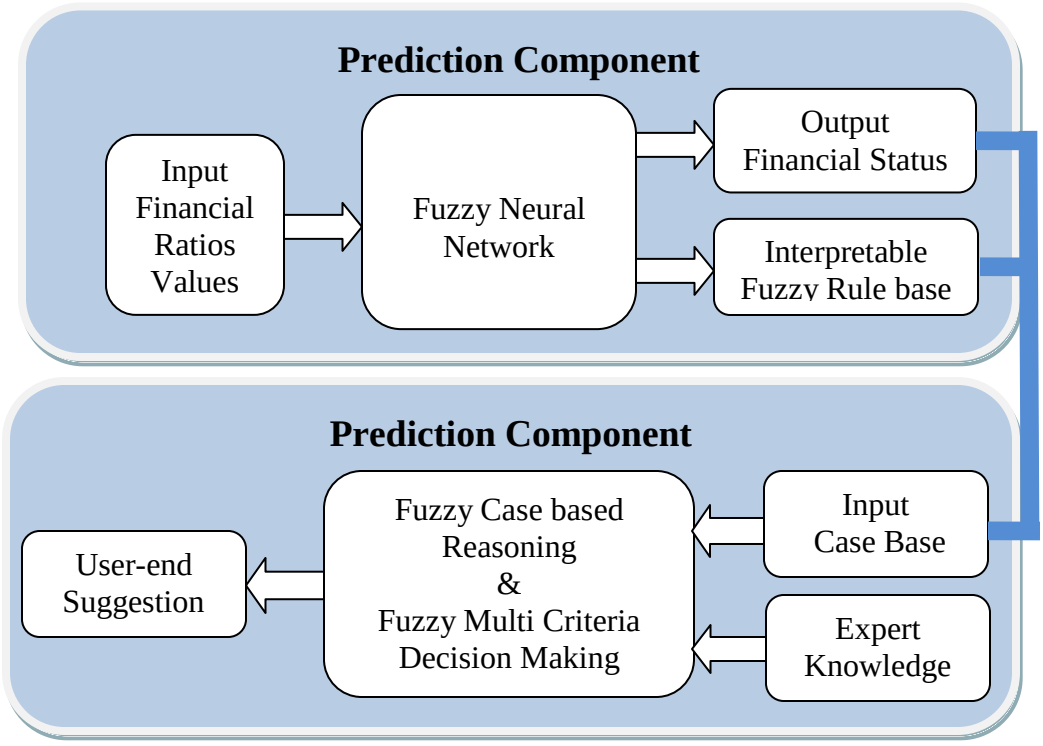


FIGURE 7.1: INTELLIGENT FINANCIAL WARNING SUPPORT SYSTEM FRAMEWORK

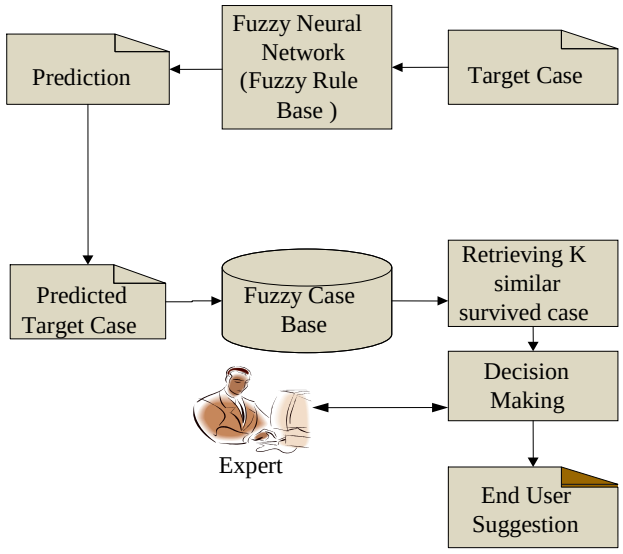


FIGURE 7.2: INTELLIGENT FINANCIAL WARNING SUPPORT SYSTEM MODEL

REFERENCES

Ahn, B.S., Cho, S.S. & Kim, C.Y. 2000, 'The integrated methodology of rough set theory and artificial neural network for business failure prediction', *Expert Systems with Applications*, vol. 18, no. 2, pp. 65-74.

Ahn, H., Lee, K. & Kim, K. 2006, 'Global optimization of support vector machines using genetic algorithms for bankruptcy prediction', in, *Neural Information Processing*, vol. 4234, Springer, Berlin, pp. 420-429.

Alam, P., Booth, D., Lee, K. & Thordarson, T. 2000, 'The use of fuzzy clustering algorithm and self-organizing neural networks for identifying potentially failing banks: an experimental study', *Expert Systems with Applications*, vol. 18, no. 3, pp. 185-199.

Alcal´a-Fdez, J., S´anchez, L., Garc´ıa, S., Del Jesus, M., Ventura, S., Garrell, J., Otero, J., Romero, C., Bacardit, J. & Rivas, V. 2009, 'KEEL: A software tool to assess evolutionary algorithms for data mining problems', *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 3, pp. 307-318.

Alcalá-Fdez, J., Herrera, F., Márquez, F. & Peregrín, A. 2007, 'Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems', *International Journal of Intelligent Systems*, vol. 22, no. 9, pp. 1035-1064.

Alfaro, E., GarcÃ-a, N., GÃ¡mez, M. & Elizondo, D. 2008, 'Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks', *Decision Support Systems*, vol. 45, no. 1, pp. 110-122.

Altman, E.I. 1968, 'Financial ratios, discriminant analysis and the prediction of corporate bankruptcy', *The Journal of Finance*, vol. 23, no. 4, pp. 589-609.

Ando, R.K. 2004, 'Exploiting unannotated corpora for tagging and chunking', *Annual Meeting of the Association for Computational Linguistics (ACL)*, Stroudsburg, USA, Article No. 13.

Ando, R.K. & Zhang, T. 2005, 'A framework for learning predictive structures from multiple tasks and unlabeled data', *The Journal of Machine Learning Research*, vol. 6, pp. 1817-1853.

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: Cybernetics*, vol. 33, no. 6, pp. 838-849.

Arnold, A., Nallapati, R. & Cohen, W.W. 2007, 'A Comparative Study of Methods for Transductive Transfer Learning', *Seventh IEEE International Conference on Data Mining Workshops*, Omaha, NE, pp. 77-82.

Atiya, A.F. 2001, 'Bankruptcy prediction for credit risk using neural networks: A survey and new results', *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 929-935.

Aue, A. & Gamon, M. 2005, 'Customizing Sentiment Classifiers to New Domains: A Case Study', *International Conference on Recent Advances in Natural Language Processing*, Borovets, Bulgaria, pp. 32-38.

Aziz, M.A. & Dar, H.A. 2006, 'Predicting corporate bankruptcy: where we stand?', *Corporate Governance*, vol. 6, no. 1, pp. 18-33.

Bahrammirzaee, A. 2010, 'A comparative survey of artificial intelligence applications in finance: Artificial neural networks, expert system and hybrid intelligent systems', *Neural Computing & Applications*, vol. 19, no. 8, pp. 1165-1195.

Balcaen, S. & Ooghe, H. 2006, '35 years of studies on business failure: An overview of the classic statistical methodologies and their related problems', *The British Accounting Review*, vol. 38, no. 1, pp. 63-93.

Balcan, M.F., Blum, A. & Srebro, N. 2008, 'A theory of learning with similarity functions', *Machine Learning*, vol. 72, no. 1, pp. 89-112.

Baralis, E., Chiusano, S. & Garza, P. 2008, 'A lazy approach to associative classification', *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 156-171.

Barandela, R., Sánchez, J.S., García, V. & Rangel, E. 2003, 'Strategies for learning in class imbalance problems', *Pattern Recognition*, vol. 36, no. 3, pp. 849-851.

Batista, G.E., Prati, R.C. & Monard, M.C. 2004, 'A study of the behavior of several methods for balancing machine learning training data', *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20-29.

Beaver, W. 1967, 'Financial ratios predictors of failure. Empirical research in accounting: Selected studies 1966', *Journal of Accounting Research*, vol. 4, pp. 71-111.

Becchetti, L. & Sierra, J. 2003, 'Bankruptcy risk and productive efficiency in manufacturing firms', *Journal of Banking & Finance*, vol. 27, no. 11, pp. 2099-2120.

Behbood, V. & Lu, J. 2011, 'Intelligent financial warning model using fuzzy neural network and case-based reasoning', *IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, Paris, France, pp. 1-6.

Behbood, V., Lu, J. & Zhang, G. 2010, 'Adaptive Inference-based learning and rule generation algorithms in fuzzy neural network for failure prediction', *IEEE International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Hangzhou, China, pp. 33-38.

Behbood, V., Lu, J. & Zhang, G. 2011, 'Long term bank failure prediction using fuzzy refinement-based transductive transfer learning', *IEEE International Conference on Fuzzy Systems*, Taipei, Taiwan, pp. 2676-2683.

Bell, T.B. 1997, 'Neural nets or the logit model? A comparison of each modelrsquos ability to predict commercial bank failures', *International Journal of Intelligent Systems in Accounting, Finance & Management*, vol. 6, no. 3, pp. 249-264.

Berenji, H.R. & Khedkar, P. 1992, 'Learning and tuning fuzzy logic controllers through reinforcements', *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 724-740.

Bezdek, J.C. & Ehrlich, R. 1984, 'FCM: The fuzzy c-means clustering algorithm', *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191-203.

Bhargava, M., Dubelaar, C. & Scott, T. 1998, 'Predicting bankruptcy in the retail sector: An examination of the validity of key measures of performance', *Journal of Retailing and Consumer Services*, vol. 5, no. 2, pp. 105-117.

Blitzer, J., Crammer, K., Kulesza, A., Pereira, F. & Wortman, J. 2007a, 'Learning bounds for domain adaptation', *Twenty-First Annual Conference on Neural Information Processing Systems*, Cambridge, MA, pp. 245-252.

Blitzer, J., Dredze, M. & Pereira, F. 2007b, 'Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification', *45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republi, pp. 440-447.

Blitzer, J., Foster, D. & Kakade, S. 2009, *Zero-shot domain adaptation: A multi-view approach*, Technical Report TTI-TR-2009-1, Toyota Technological Institute Chicago.

Blitzer, J., McOnald, R. & Pereira, F. 2006, 'Domain adaptation with structural correspondence learning', *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, pp. 120-128.

Blum, A. & Mitchell, T. 1998, 'Combining labeled and unlabeled data with co-training', *Eleventh Annual Conference on Computational Learning Theory*, Madison, WI, pp. 92-100.

Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B. & Smola, A.J. 2006, 'Integrating structured biological data by kernel maximum mean discrepancy', *Bioinformatics*, vol. 22, no. 14, pp. 49-57.

Boyacioglu, M.A., Kara, Y. & Baykan, Ö.K. 2009, 'Predicting bank financial failures using neural networks, support vector machines and multivariate statistical methods: A comparative analysis in the sample of savings deposit insurance fund (SDIF) transferred banks in Turkey', *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3355-3366.

Canbas, S., Cabuk, A. & Kilic, S.B. 2005, 'Prediction of commercial bank failure via multivariate statistical analysis of financial structures: The Turkish case', *European Journal of Operational Research*, vol. 166, no. 2, pp. 528-546.

Caruana, R. & Niculescu-Mizil, A. 2006, 'An empirical comparison of supervised learning algorithms', *23rd International Conference on Machine Learning*, Pittsburgh, PA, pp. 161-168.

Chan, A.P.F., Ng, W.W.Y., Yeung, D.S., Tsang, E.C.C. & Firth, M. 2006, 'Bankruptcy prediction using multiple classifier system with mutual information feature grouping', *IEEE International Conference on Systems, Man and Cybernetics*, Taipei, Taiwan, pp. 845-850.

Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, P. 2002, 'SMOTE: Synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357.

Chawla, N., Lazarevic, A., Hall, L. & Bowyer, K. 2003, 'SMOTEBoost: Improving prediction of the minority class in boosting', *7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Cavtat-Dubrovink, Croatia, pp. 107-119.

Chawla, N.V., Japkowicz, N. & Kotcz, A. 2004, 'Editorial: Special issue on learning from imbalanced data sets', *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 1-6.

Chawla, N.V. & Karakoulas, G. 2005, 'Learning from labeled and unlabeled data: An empirical study across techniques and domains', *Journal of Artificial Intelligence Research*, vol. 23, no. 1, pp. 331-366.

Chen, B., Lam, W., Tsang, I. & Wong, T.L. 2009a, 'Extracting discriminative concepts for domain adaptation in text mining', *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, pp. 179-188.

Chen, H.-J., Huang, S.Y. & Lin, C.-S. 2009b, 'Alternative diagnosis of corporate bankruptcy: A neuro fuzzy approach', *Expert Systems with Applications*, vol. 36, no. 4, pp. 7710-7720.

Chen, L.H. & Hsiao, H.D. 2008, 'Feature selection to diagnose a business crisis by using a real GA-based support vector machine: An empirical study', *Expert Systems with Applications*, vol. 35, no. 3, pp. 1145-1155.

Chiang, D.-A. & Lin, N.P. 1999, 'Correlation of fuzzy sets', *Fuzzy Sets and Systems*, vol. 102, no. 2, pp. 221-226.

Ciaramita, M. & Chapelle, O. 2010, 'Adaptive parameters for entity recognition with perceptron HMMs', *Workshop on Domain Adaptation for Natural Language Processing*, Uppsala, Sweden, pp. 1-7.

Clark, P. & Niblett, T. 1989, 'The CN2 induction algorithm', *Machine Learning*, vol. 3, no. 4, pp. 261-283.

Cole, R.A. & Gunther, J. 1995, *A CAMEL Rating's Shelf Life*, Federal Reserve Bank of Dallas, Financial Industry Studies.

Cole, R.A. & Gunther, J.W. 1998, 'Predicting bank failures: A comparison of on-and off-site monitoring systems', *Journal of Financial Services Research*, vol. 13, no. 2, pp. 103-117.

Cominetti, O., Matzavinos, A., Samarasinghe, S., Kulasiri, D., Liu, S., Maini, P.K. & Erban, R. 2010, 'DifFUZZY: A fuzzy clustering algorithm for complex datasets', *International Journal of Computational Intelligence in Bioinformatics and Systems Biology*, vol. 1, no. 4, pp. 402-417.

Conover, W.J. 1999, *Practical Nonparametric Statistics*, John Wiley and Sons, Oxford.

Cortes, C., Mohri, M., Riley, M. & Rostamizadeh, A. 2008, 'Sample selection bias correction theory', *19th International Conference on Algorithmic Learning Theory*, Budapest, Hungary, pp. 38-53.

Cortes, E.A., MartÃnez, M.G. & Rubio, N.G. 2007, 'A boosting approach for corporate failure prediction', *Applied Intelligence*, vol. 27, no. 1, pp. 29-37.

Dai, W., Xue, G.-R., Yang, Q. & Yu, Y. 2007a, 'Co-clustering based classification for out-of-domain documents', *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*,, San Jose, CA, pp. 210-219.

Dai, W., Xue, G.R., Yang, Q. & Yu, Y. 2007b, 'Transferring naive bayes classifiers for text classification', *22nd National Conference on Artificial Intelligence*, Vancouver, Canada, pp. 540-545.

Davis, E.P. & Karim, D. 2008, 'Comparing early warning systems for banking crises', *Journal of Financial Stability*, vol. 4, no. 2, pp. 89-120.

Demsar, J. 2006, 'Statistical comparisons of classifiers over multiple data sets', *The Journal of Machine Learning Research*, vol. 7, pp. 1-30.

Demyanyk, Y. & Hasan, I. 2010, 'Financial crises and bank failures: A review of prediction methods', *Omega*, vol. 38, no. 5, pp. 315-324.

Ding, Y., Song, X. & Zen, Y. 2008, 'Forecasting financial condition of Chinese listed companies based on support vector machine', *Expert Systems with Applications*, vol. 34, no. 4, pp. 3081-3089.

Donoher, W.J. 2004, 'To file or not to file? Systemic incentives, corporate control, and the bankruptcy decision', *Journal of Management*, vol. 30, no. 2, pp. 239-262.

Fabling, R. & Grimes, A., 'Insolvency and economic development: Regional variation and adjustment', *Journal of Economics and Business*, vol. 57, no. 4, pp. 339-359.

Fan, W., Stolfo, S.J., Zhang, J. & Chan, P.K. 1999, 'AdaCost: Misclassification cost-sensitive boosting', *16th International Conference on Machine Learning*, Bled, Slovenia, pp. 97-105.

Fernández, A., del Jesus, M.J. & Herrera, F. 2009a, 'Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets', *International Journal of Approximate Reasoning*, vol. 50, no. 3, pp. 561-577.

Fernández, A., del Jesus, M.J. & Herrera, F. 2009b, 'On the influence of an adaptive inference system in fuzzy rule based classification systems for imbalanced data-sets', *Expert Systems with Applications*, vol. 36, no. 6, pp. 9805-9812.

Fernández, A., García, S., del Jesus, M.J. & Herrera, F. 2008, 'A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets', *Fuzzy Sets and Systems*, vol. 159, no. 18, pp. 2378-2398.

Fletcher, D. & Goss, E. 1993, 'Forecasting with neural networks : An application using bankruptcy data', *Information & Management*, vol. 24, no. 3, pp. 159-167.

Freund, Y. & Schapire, R. 1995, 'A Desicion-theoretic generalization of on-line learning and an application to boosting', *Second European Conference on Computational Learning Theory*, Barcelona, Spain, pp. 23-37.

Freund, Y. & Schapire, R.E. 1996, 'Experiments with a new boosting algorithm', *International Conference on Machine Learning*, Bar, Italy, pp. 148-156.

Frydman, H., Altman, E.I. & Kao, D.-L. 1985, 'Introducing recursive partitioning for financial classification: The case of financial distress', *The Journal of Finance*, vol. 40, no. 1, pp. 269-291.

Fung, G.P.C., Yu, J.X., Hongjun, L. & Yu, P.S. 2006, 'Text classification without negative examples revisit', *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 6-20.

Gallupe, R.B. 2007, 'The tyranny of methodologies in information systems research', *SIGMIS Database*, vol. 38, no. 3, pp. 20-28.

Gao, J., Fan, W., Jiang, J. & Han, J. 2008, 'Knowledge transfer via multiple model local structure mapping', *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, USA, pp. 283-291.

Garcia, S., Fernandez, A., Luengo, J. & Herrera, F. 2010, 'Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power', *Information Sciences*, vol. 180, no. 10, pp. 2044-2064.

Garcia, S. & Herrera, F. 2008, 'An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons', *Journal of Machine Learning Research*, vol. 9, no. 2677-2694, p. 66.

Gepp, A., Kumar, K. & Bhattacharya, S. 2010, 'Business failure prediction using decision trees', *Journal of Forecasting*, vol. 29, no. 6, pp. 536-555.

Ghahramani, Z. & Jordan, M.I. 1995, *Learning from Incomplete Data*, MIT Press, Cambridge, MA.

Gorzalczany, M.B. & Piasta, Z. 1999, 'Neuro-fuzzy approach versus rough-set inspired methodology for intelligent decision support', *Information Sciences*, vol. 120, no. 1-4, pp. 45-68.

Gowda, K. & Krishna, G. 1979, 'The condensed nearest neighbor rule using the concept of mutual nearest neighborhood', *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 488-490.

Haibo, H. & Garcia, E.A. 2009, 'Learning from Imbalanced Data', *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284.

Heckman, J.J. 1977, *Sample Selection bias as a Specification Error with an Application to the Estimation of Labor Supply Functions*, National Bureau of Economic Research Cambridge, Mass., USA.

Holm, S. 1979, 'A simple sequentially rejective multiple test procedure', *Scandinavian Journal of Statistics*, vol. 6, pp. 65-70.

Hosmer, D.W. & Lemeshow, S. 1989, *Applied Logistic Regression*, Wiley, New York.

Hua, Z., Wang, Y., Xu, X., Zhang, B. & Liang, L. 2007, 'Predicting corporate financial distress based on integration of support vector machine and logistic regression', *Expert Systems with Applications*, vol. 33, no. 2, pp. 434-440.

Huang, D.T., Chang, B. & Liu, Z.C. 2012, 'Bank failure prediction models: for the developing and developed countries', *Quality & Quantity*, vol. 46, pp. 553-558.

Huang, F. & Yates, A. 2009, 'Distributional representations for handling sparsity in supervised sequence-labeling', *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International*

Conference on Natural Language Processing, Singapore, Singapore, pp. 495-503.

Huang, F. & Yates, A. 2010a, 'Exploring representation-learning approaches to domain adaptation', *Workshop on Domain Adaptation for Natural Language Processing*, Uppsala, Sweden, pp. 23-30.

Huang, F. & Yates, A. 2010b, 'Open-domain semantic role labeling by modeling word spans', *48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, pp. 968-978.

Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M. & Scholkopf, B. 2007, 'Correcting sample selection bias by unlabeled data', *Advances in Neural Information Processing Systems*, vol. 19, pp. 601-608.

Huarng, K., Yu, H. & Chen, C. 2005, 'The application of decision trees to forecast financial distressed companies', *International Conference on Intelligent Technologies and Applied Statistics*, Taipei, Taiwan, pp. 126-131.

Huysmans, J., Baesens, B., Vanthienen, J. & van Gestel, T. 2006, 'Failure prediction with self organizing maps', *Expert Systems with Applications*, vol. 30, no. 3, pp. 479-487.

Jang, J.S.R. 1993, 'ANFIS: Adaptive-network-based fuzzy inference system', *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665-685.

Japkowicz, N. 2001, 'Supervised versus unsupervised binary-learning by feedforward neural networks', *Machine Learning*, vol. 42, no. 1, pp. 97-122.

Japkowicz, N. & Stephen, S. 2002, 'The class imbalance problem: A systematic study', *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429-450.

Jeong, M., Lin, C.Y. & Lee, G.G. 2009, 'Semi-supervised speech act recognition in emails and forums', *Conference on Empirical Methods in Natural Language Processing*, Singapore, Singapore, pp. 1250-1259.

Ji, Y.S., Chen, J.J., Niu, G., Shang, L. & Dai, X.Y. 2011, 'Transfer learning via multi-view principal component analysis', *Journal of Computer Science and Technology*, vol. 26, no. 1, pp. 81-98.

Jiang, J. & Zhai, C.X. 2007a, 'Instance weighting for domain adaptation in NLP', *45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 264-271.

Jiang, J. & Zhai, C.X. 2007b, 'A Two-stage approach to domain adaptation for statistical classifiers', *Sixteenth ACM Conference on Information and Knowledge Management*, Lisbon, Portugal, pp. 401-410.

Joachims, T. 1999a, 'Making large-scale support vector machine learning practical', in, *Advances in Kernel Methods*, MIT Press, Cambridge, MA, pp. 169-184.

Joachims, T. 1999b, 'Transductive inference for text classification using support vector machines', *Sixteenth International Conference on Machine Learning*, Bled, Slovenia, pp. 200-209.

Joos, P., Vanhoof, K., Ooghe, H. & Sierens, N. 1998, 'Credit classification: A comparison of logit models and decision trees', *ECML Workshop on Applications of Machine Learning and Data Mining in Finance*, Chemnitz, Germany, pp. 59-72.

Joshi, M.V., Kumar, V. & Agarwal, R.C. 2001, 'Evaluating boosting algorithms to classify rare classes: Comparison and improvements', *IEEE International Conference on Data Mining*, San Jose, CA, pp. 257-264.

Kasabov, N.K. & Qun, S. 2002, 'DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction', *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144-154.

Khotanzad, A., Elragal, H. & Lu, T.L. 2000, 'Combination of artificial neural-network forecasters for prediction of natural gas consumption', *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 464-473.

Kim, T.Y., Oh, K.J., Sohn, I. & Hwang, C. 2004, 'Usefulness of artificial neural networks for early warning system of economic crisis', *Expert Systems with Applications*, vol. 26, no. 4, pp. 583-590.

Klinkenberg, R. & Joachims, T. 2000, 'Detecting concept drift with support vector machines', *Seventeenth International Conference on Machine Learning*, Stanford, CA, pp. 487 - 494

Kotsiantis, S., Kanellopoulos, D. & Pintelas, P. 2006, 'Handling imbalanced datasets: A review', *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25-36.

Kubat, M., Holte, R.C. & Matwin, S. 1998, 'Machine learning for the detection of oil spills in satellite radar images', *Machine Learning*, vol. 30, no. 2, pp. 195-215.

Kubat, M. & Matwin, S. 1997, 'Addressing The Curse of Imbalanced Training Sets: One-sided Selection', *14th International Conference on Machine Learning*, San Francisco, CA, pp. 179-186.

Kumar, P.R. & Ravi, V. 2006, 'Bankruptcy prediction in banks by fuzzy rule based classifier', *1st International Conference on Digital Information Management*, Bangalore, India, pp. 222-227.

Kuncheva, L.I. & Rodriguez, J.J. 2007, 'Classifier ensembles with a random linear oracle', *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 4, pp. 500-508.

Laitinen, E.K. & Laitinen, T. 2000, 'Bankruptcy prediction: Application of the Taylor's expansion in logistic regression', *International Review of Financial Analysis*, vol. 9, no. 4, pp. 327-349.

Lane, W.R., Looney, S.W. & Wansley, J.W. 1986, 'An application of the cox proportional hazards model to bank failure', *Journal of Banking & Finance*, vol. 10, no. 4, pp. 511-531.

Lee, C.H., Quek, C. & Maskell, D.L. 2006, 'A brain inspired fuzzy neuro-predictor for bank failure analysis', *IEEE Congress on Evolutionary Computation*, Vancouver, Canada, pp. 2163-2170.

Lee, K., Booth, D. & Alam, P. 2005, 'A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms', *Expert Systems with Applications*, vol. 29, no. 1, pp. 1-16.

Lewis, D. & Gale, W. 1994, 'Training text classifiers by uncertainty sampling', *Seventeenth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, pp. 3-12.

Li, H. & Sun, J. 2008, 'Ranking-order case-based reasoning for financial distress prediction', *Knowledge-Based Systems*, vol. 21, no. 8, pp. 868-878.
Li, H. & Sun, J. 2009a, 'Forecasting business failure in China using hybrid case-based reasoning', *Journal of Forecasting*, vol. 9999, no. 9999, p. n/a.
Li, H. & Sun, J. 2009b, 'Gaussian case-based reasoning for business failure prediction with empirical data in China', *Information Sciences*, vol. 179, no. 1-2, pp. 89-108.
Li, H. & Sun, J. 2009c, 'Hybridizing principles of the Electre method with case-based reasoning for data mining: Electre-CBR-I and Electre-CBR-II', *European Journal of Operational Research*, vol. 197, no. 1, pp. 214-224.
Li, H. & Sun, J. 2010, 'Business failure prediction using hybrid2 case-based reasoning (H2CBR)', *Computers & Operations Research*, vol. 37, no. 1, pp. 137-151.
Li, S.-T. & Ho, H.-F. 2009, 'Predicting financial activity with evolutionary fuzzy case-based reasoning', *Expert Systems with Applications*, vol. 36, no. 1, pp. 411-422.
Lin, C.-J. & Lin, C.-T. 1997, 'An ART-based fuzzy adaptive learning control network', *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 4, pp. 477-496.
Lin, C.-S., Khan, H.A., Chang, R.-Y. & Wang, Y.-C. 2008, 'A new approach to modeling early warning systems for currency crises: Can a machine-learning fuzzy expert system predict the currency crises effectively?', *Journal of International Money and Finance*, vol. 27, no. 7, pp. 1098-1121.
Lin, C.-T. & Lee, C.S.G. 1996, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall, Englewood Cliffs, NJ.
Ling, C.X., Yang, Q., Wang, J. & Zhang, S. 2004, 'Decision trees with minimal costs', *21th International Conference on Machine Learning*, Banff, Canada, pp. 69-45.
Ling, X., Dai, W., Xue, G.-R., Yang, Q. & Yu, Y. 2008a, 'Spectral domain-transfer learning', *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, NV, pp. 488-496.
Ling, X., Xue, G.R., Dai, W., Jiang, Y., Yang, Q. & Yu, Y. 2008b, 'Can chinese web pages be classified with english data source?', *17th International Conference on World Wide Web*, Beijing, China, pp. 969-978.
Lu, J., Tokinaga, S. & Ikeda, Y. 2006, 'Explanatory rule extraction based on the trained neural network and the genetic programming', *Journal of the Operations Research Society of Japan-Keiei Kagaku*, vol. 49, no. 1, p. 66.
Lu, J., Zhang, G., Ruan, D. & Wu, F. 2007, *Multi-Objective Group Decision Making: Methods, Software and Applications with Fuzzy Set Techniques*, Imperial College Press, London.
Macqueen, J.B. 1967, 'Some methods of classification and analysis of multivariate observations', *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, pp. 281-297.
Maddala, G.S. 1977, *Econometrics*, McGraw-Hill, New York.
Marais, M.L., Patell, J.M. & Wolfson, M.A. 1984, 'The experimental design of classification models: An application of recursive partitioning and

bootstrapping to commercial bank loan classifications', *Journal of Accounting Research*, vol. 22, pp. 87-114.

Margineantu, D. 2002, 'Class probability estimation and cost-sensitive classification decisions', *European Conference on Machine Learning*, Helsinki, Finland, pp. 270-281.

Margolis, A. 2011, *A Literature Review of Domain Adaptation with Unlabeled Data*, University of Washington, Washington.

Margolis, A., Livescu, K. & Ostendorf, M. 2010, 'Domain adaptation with unlabeled data for dialog act tagging', *Workshop on Domain Adaptation for Natural Language Processing*, Uppsala, Sweden, pp. 45-52.

Marquez, F.A., Peregrin, A. & Herrera, F. 2007, 'Cooperative evolutionary learning of linguistic fuzzy rules and parametric aggregation connectors for Mamdani fuzzy systems', *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1162-1178.

Martin, D. 1977, 'Early warning of bank failure : A logit regression approach', *Journal of Banking & Finance*, vol. 1, no. 3, pp. 249-276.

McClosky, D., Charniak, E. & Johnson, M. 2006, 'Reranking and self-training for parser adaptation', *21st International Conference on Computational Linguistics*, Sydney, Australia, pp. 337-344.

Mcleay, S. & Omar, A. 2000, 'The sensitivity of prediction models to the non-normality of bounded and unbounded financial ratios', *The British Accounting Review*, vol. 32, no. 2, pp. 213-230.

Min, J.H. & Jeong, C. 2009, 'A binary classification method for bankruptcy prediction', *Expert Systems with Applications*, vol. 36, no. 3, Part 1, pp. 5256-5263.

Min, J.H. & Lee, Y.-C. 2005, 'Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters', *Expert Systems with Applications*, vol. 28, no. 4, pp. 603-614.

Min, S.H., Lee, J. & Han, I. 2006, 'Hybrid genetic algorithms and support vector machines for bankruptcy prediction', *Expert Systems with Applications*, vol. 31, no. 3, pp. 652-660.

Moreno-Torres, J.G., Raeder, T., Alaiz-Rodriguez, R., Chawla, N.V. & Herrera, F. 2012, 'A unifying view on dataset shift in classification', *Pattern Recognition*, vol. 45, no. 1, pp. 521-530.

Moses, D. & Liao, S.S. 1987, 'On developing models for failure prediction', *Journal of Commercial Bank Lending*, vol. 69, pp. 27-38.

Nauck, D., Klawonn, F. & Kruse, R. 1997, *Foundations of Neuro-Fuzzy Systems*, John Wiley & Sons, Inc., New York.

Ng, G.S., Quek, C. & Jiang, H. 2008, 'FCMAC-EWS: A bank failure early warning system based on a novel localized pattern learning and semantically associative fuzzy neural network', *Expert Systems with Applications*, vol. 34, no. 2, pp. 989-1003.

Nguyen, M.N., Shi, D. & Quek, C. 2008, 'A nature inspired Ying-Yang approach for intelligent decision support in bank solvency analysis', *Expert Systems with Applications*, vol. 34, no. 4, pp. 2576-2587.

Nigam, K., McCallum, A.K., Thrun, S. & Mitchell, T. 2000, 'Text classification from labeled and unlabeled documents using EM', *Machine Learning*, vol. 39, no. 2-3, pp. 103-134.

Niu, L., Lu, J. & Zhang, G. 2009, *Cognition-Driven Decision Support for Business Intelligence: Models, Techniques, Systems and Applications*, Springer, Berlin.

Odom, M.D. & Sharda, R. 1990, 'A neural network model for bankruptcy prediction', *International Joint Conference on Neural Networks*, Washington, DC, pp. 163-168.

Oentaryo, R.J., Pasquier, M. & Quek, C. 2008, 'GenSoFNN-Yager: A novel brain-inspired generic self-organizing neuro-fuzzy system realizing Yager inference', *Expert Systems with Applications*, vol. 35, no. 4, pp. 1825-1840.

Orriols-Puig, A. & Bernadó-Mansilla, E. 2009, 'Evolutionary rule-based systems for imbalanced data sets', *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 3, pp. 213-225.

Page, L., Brin, S., Motwani, R. & Winograd, T. 1998, 'The page rank citation ranking: Bringing order to the Web', *7th International World Wide Web Conference*, Brisbane, Australia, pp. 161-172.

Pan, S.J., Kwok, J.T. & Yang, Q. 2008, 'Transfer learning via dimensionality reduction', *23th National Conference on Artificial intelligence*, Chicago, IL, pp. 677-682.

Pan, S.J., Ni, X., Sun, J.T., Yang, Q. & Chen, Z. 2010, 'Cross-domain sentiment classification via spectral feature alignment', *19th International Conference on World wide web*, Raleigh, NC, pp. 751-760.

Pan, S.J., Tsang, I.W., Kwok, J.T. & Yang, Q. 2009, 'Domain adaptation via transfer component analysis', *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199-210.

Pan, S.J. & Yang, Q. 2010, 'A survey on transfer learning', *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359.

Park, C.-S. & Han, I. 2002, 'A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction', *Expert Systems with Applications*, vol. 23, no. 3, pp. 255-264.

Pasquier, M., Quek, C. & Toh, M. 2001, 'Fuzzylot: A novel self-organising fuzzy-neural rule-based pilot system for automated vehicles', *Neural Networks*, vol. 14, no. 8, pp. 1099-1112.

Pérez, Ó. & Sánchez-Montañés, M. 2007, 'A new learning strategy for classification problems with different training and test distributions', *Computational and Ambient Intelligence*, pp. 178-185.

Peymanfar, A., Khoei, A. & Hadidi, K. 2007, 'A new ANFIS based learning algorithm for CMOS neuro-fuzzy controllers', *14th IEEE International Conference on Electronics, Circuits and Systems*, Marrakech, Morocco, pp. 890-893.

Piramuthu, S., Ragavan, H. & Shaw, M.J. 1998, 'Using feature construction to improve the performance of neural networks', *Management Science*, pp. 416-430.

Prettenhofer, P. & Stein, B. 2010, 'Cross-language text classification using structural correspondence learning', *48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, pp. 1118-1127.

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: Cybernetics*, vol. 29, no. 6, pp. 859-870.

Quek, C., Zhou, R.W. & Lee, C.H. 2009, 'A novel fuzzy neural approach to data reconstruction and failure prediction', *International Journal of Intelligent Systems in Accounting and Finance Management*, vol. 16, no. 2, pp. 165-187.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A. & Lawrence, N. 2009, *Dataset Shift in Machine Learning*, The MIT Press, Cambridge.

Ramze Rezaee, M., Goedhart, B., Lelieveldt, B. & Reiber, J. 1999, 'Fuzzy feature selection', *Pattern Recognition*, vol. 32, no. 12, pp. 2011-2019.

Raskutti, B. & Kowalczyk, A. 2004, 'Extreme re-balancing for SVMs: A case study', *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 60-69.

Ravi Kumar, P. & Ravi, V. 2007, 'Bankruptcy prediction in banks and firms via statistical and intelligent techniques: A review', *European Journal of Operational Research*, vol. 180, no. 1, pp. 1-28.

Ravi, V., Kurniawan, H., Thai, P.N.K. & Kumar, P.R. 2008, 'Soft computing system for bank performance prediction', *Applied Soft Computing*, vol. 8, no. 1, pp. 305-315.

Ravi, V. & Pramodh, C. 2008, 'Threshold accepting trained principal component neural network and feature subset selection: Application to bankruptcy prediction in banks', *Applied Soft Computing*, vol. 8, no. 4, pp. 1539-1548.

Ravikumar, P. & Ravi, V. 2006, 'Bankruptcy prediction in banks by an ensemble classifier', *IEEE International Conference on Industrial Technology*, Mumbai, India, pp. 2032-2036.

Ren, J., Shi, X., Fan, W. & Yu, P.S. 2008, 'Type-independent correction of sample selection bias via structural discovery and re-balancing', *SIAM International Conference on Data Mining*, Atlanta, GA, pp. 565–576.

Rezaee, M.R., Goedhart, B., Lelieveldt, B. & Reiber, J. 1999, 'Fuzzy feature selection', *Pattern Recognition*, vol. 32, no. 12, pp. 2011-2019.

Rhee, F.C.H. & Lee, Y.J. 1999, 'Unsupervised feature selection using a fuzzy-genetic algorithm', *IEEE International Conference on Fuzzy Systems*, Seoul, Korea, pp. 1266-1269.

Rigutini, L., Maggini, M. & Liu, B. 2005, 'An EM based training algorithm for cross-language text categorization', *IEEE/WIC/ACM International Conference on Web Intelligence*, Compiegne, France, pp. 529-535.

Roark, B. & Bacchiani, M. 2003, 'Supervised and unsupervised PCFG adaptation to novel domains', *Conference of the North American Chapter of the Association*

for Computational Linguistics on Human Language, Edmonton, Canada, pp. 126-133.

Rosset, S., Zhu, J., Zou, H. & Hastie, T. 2005, 'A method for inferring label sampling mechanisms in semi-supervised learning', *Ann Arbor*, vol. 1001, p. 48109.

Saerens, M., Latinne, P. & Decaestecker, C. 2002, 'Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure', *Neural Computation*, vol. 14, no. 1, pp. 21-41.

Sagae, K. 2010, 'Self-training without reranking for parser domain adaptation and its impact on semantic role labeling', *Workshop on Domain Adaptation for Natural Language Processing*, Uppsala, Sweden, pp. 37-44.

Sagae, K. & Tsujii, J. 2007, 'Dependency parsing and domain adaptation with LR models and parser ensembles', *Eleventh Conference on Computational Natural Language Learning*, Prague, Czech Republic, pp. 1044-1050.

Salchenberger, L.M., Cinar, E.M. & Lash, N.A. 1992, 'Neural networks: A new tool for predicting thrift failures', *Decision Sciences*, vol. 23, no. 4, pp. 899-916.

Sandu, O., Carenini, G., Murray, G. & Ng, R. 2010, 'Domain adaptation to summarize human conversations', *Workshop on Domain Adaptation for Natural Language Processing*, Uppsala, Sweden, pp. 16-22.

Satpal, S. & Sarawagi, S. 2007, 'Domain adaptation of conditional probability models via feature subsetting', *11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Warsaw, Pland, pp. 224-235.

Shi, L., Mihalcea, R. & Tian, M. 2010, 'Cross language text classification by model translation and semi-supervised learning', *Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, pp. 1057-1067.

Shin, K.-S., Lee, T.S. & Kim, H.-J. 2005, 'An application of support vector machines in bankruptcy prediction model', *Expert Systems with Applications*, vol. 28, no. 1, pp. 127-135.

Shin, K.-S. & Lee, Y.-J. 2002, 'A genetic algorithm application in bankruptcy prediction modeling', *Expert Systems with Applications*, vol. 23, no. 3, pp. 321-328.

Sim, J., Tung, W.L. & Chai, Q. 2006a, 'FCMAC-Yager: a novel Yager-iInference-scheme-based fuzzy CMAC', *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1394-1410.

Sim, J., Tung, W.L. & Chai, Q. 2006b, 'FCMAC-Yager: A Novel Yager-Inference-Scheme-Based Fuzzy CMAC', *Neural Networks, IEEE Transactions on*, vol. 17, no. 6, pp. 1394-1410.

Singh, A., Quek, C. & Cho, S.Y. 2008, 'DCT-Yager FNN: A novel Yager-based fuzzy neural network with the discrete clustering technique', *IEEE Transactions on Neural Networks*, vol. 19, no. 4, pp. 625-644.

Spanos, M., Dounias, G., Matsatsinis, N. & Zopounidis, C. 1999, 'A fuzzy knowledge-based decision aiding method for the assessment of financial risks: The case of corporate bankruptcy prediction', *European Symposium on Intelligent Techniques*, Aachen, Germany, pp. 14-21.

Sugiyama, M., Nakajima, S., Kashima, H., Von Buenau, P. & Kawanabe, M. 2008, 'Direct importance estimation with model selection and its application to covariate shift adaptation', *Advances in Neural Information Processing Systems*, vol. 20, pp. 1433-1440.

Sun, Y., Wong, A.K.C. & Kamel, M.S. 2009, 'Classification of imbalanced data: A review', *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687-719.

Swicegood, P. & Clark, J.A. 2001, 'Off-site monitoring systems for predicting bank underperformance: A comparison of neural networks, discriminant analysis, and professional human judgment', *International Journal of Intelligent Systems in Accounting, Finance & Management*, vol. 10, no. 3, pp. 169-186.

Tam, K.Y. 1991, 'Neural network models and the prediction of bank bankruptcy', *Omega*, vol. 19, no. 5, pp. 429-445.

Tam, K.Y. & Kiang, M.Y. 1992, 'Managerial applications of neural networks: The case of bank failure predictions', *Management Science*, vol. 38, no. 7, pp. 926-947.

Tamari, M. 1966, 'Financial ratios as a means of forecasting bankruptcy', *Management International Review*, vol. 4, pp. 15-21.

Tan, S. & Cheng, X. 2009, 'Improving SCL model for sentiment-transfer learning', *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, CO, pp. 181-184.

Tan, S., Cheng, X., Wang, Y. & Xu, H. 2009, 'Adapting naive bayes to domain adaptation for sentiment analysis', *31th European Conference on Advances in Information Retrieval*, Toulouse, France, pp. 337-349.

Tan, S., Wang, Y., Wu, G. & Cheng, X. 2008, 'Using unlabeled data to handle domain-transfer problem of semantic detection', *ACM Symposium on Applied Computing*, Fortaleza, Brazil, pp. 896-903.

Tang, T.-C. & Chi, L.-C. 2005, 'Predicting multilateral trade credit risks: Comparisons of Logit and fuzzy logic models using ROC curve analysis', *Expert Systems with Applications*, vol. 28, no. 3, pp. 547-556.

Theil, H. 1971, *Principles of Econometrics*, J. Wiley and Sons, New York.

Ting, K.M. 2000, 'A comparative study of cost-sensitive boosting algorithms', *Seventeenth International Conference on Machine Learning*, Stanford, CA, pp. 983 - 990.

Tishby, N., Pereira, F.C. & Bialek, W. 2000, 'The information bottleneck method', *37th Annual Allerton Conference on Communication, Control and Computing*, Champaign, IL, pp. 368-377.

Tomek, I. 1976, 'Two modifications of CNN', *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, pp. 769-772.

Tsai, C.F. & Wu, J.W. 2008, 'Using neural network ensembles for bankruptcy prediction and credit scoring', *Expert Systems with Applications*, vol. 34, no. 4, pp. 2639-2649.

Tsuboi, Y., Kashima, H., Hido, S., Bickel, S. & Sugiyama, M. 2009, 'Direct density ratio estimation for large-scale covariate shift adaptation', *Information and Media Technologies*, vol. 4, no. 2, pp. 529-546.

Tung, W.L. & Quek, C. 2002, 'GenSoFNN: A generic self-organizing fuzzy neural network', *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1075-1086.

Tung, W.L. & Quek, C. 2004, 'Falcon: Neural fuzzy control and decision systems using FKP and PFKP clustering algorithms', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 686-695.

Tung, W.L., Quek, C. & Cheng, P. 2004, 'GenSo-EWS: A novel neural-fuzzy based early warning system for predicting bank failures', *Neural Networks*, vol. 17, no. 4, pp. 567-587.

Vaishnavi, V. & Kuechler, W. 2009, *Design Research in Information Systems*, < http://ais.affiniscape.com/displaycommon.cfm?an=1&subarticlenbr=279 >.

Verikas, A., Kalsyte, Z., Bacauskiene, M. & Gelzinis, A. 2010, 'Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: A survey', *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 14, no. 9, pp. 995-1010.

Vigier, H.P. & Terceño, A. 2008, 'A model for the prediction of "diseases" of firms by means of fuzzy relations', *Fuzzy Sets and Systems*, vol. 159, no. 17, pp. 2299-2316.

Wan, X. 2009, 'Co-training for Cross-lingual Sentiment Classification', *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*, Suntec, Singapore, pp. 235-243.

Wang, C. & Mahadevan, S. 2009, 'A general framework for manifold alignment', *AAAI Fall Symposium on Manifold Learning and its Applications*, Arlington, VA, pp. 53-58.

Wang, C. & Mahadevan, S. 2011, 'Heterogeneous domain adaptation using manifold alignment', *22th International Joint Conference on Artificial Intelligence*, Barcelona, Spain, pp. 1541-1546.

Wang, L., Sugiyama, M., Yang, C., Hatano, K. & Feng, J. 2008, 'Theory and algorithm for learning with dissimilarity functions', *Neural Computation*, vol. 21, no. 5, pp. 1459-1484.

Wang, L., Yang, C. & Feng, J. 2007, 'On learning with dissimilarity functions', *International Conference on Machine Learning*, Corvallis, OR, pp. 991-998.

Wang, W. 2009, 'Combining discriminative re-ranking and co-training for parsing mandarin speech transcripts', *IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, pp. 4705-4708.

Wei, B. & Pal, C. 2010, 'Cross lingual adaptation: An experiment on sentiment classifications', *Conference on Association for Computational Linguistics*, Uppsala, Sweden, pp. 258-262.

Weiss, G.M. 2004, 'Mining with rarity: A unifying framework', *SIGKDD Exploration Newsletter*, vol. 6, no. 1, pp. 7-19.

Wilson, R.L. & Sharda, R. 1994, 'Bankruptcy prediction using neural networks', *Decision Support Systems*, vol. 11, no. 5, pp. 545-557.

Wu, G. & Chang, E.Y. 2003, 'Class-boundary alignment for imbalanced dataset learning', *International Conference on Machine Learning: Workshop on Learning from Imbalanced Data Sets*, Washington, DC, pp. 49-56.

Xing, D., Dai, W., Xue, G.-R. & Yu, Y. 2007, 'Bridged refinement for transfer learning', *11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Warsaw, Poland, pp. 324-335.

Xue, G.-R., Dai, W., Yang, Q. & Yu, Y. 2008, 'Topic-bridged PLSA for cross-domain text classification', *31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, Singapore, pp. 627-634.

YANG, Q. 2006, '10 CHALLENGING PROBLEMS IN DATA MINING RESEARCH', *International Journal of Information Technology & Decision Making* vol. 5, no. 4, pp. 597-604.

Yang, Q. 2009, 'Transfer learning beyond text classification', *1st Asian Conference on Machine Learning: Advances in Machine Learning*, Nanjing, China, pp. 10 - 22

Yeung, D.S., Ng, W.W.Y., Chan, A.P.F., Chan, P.P.K., Firth, M. & Tsang, E.C.C. 2007a, 'Bankruptcy prediction using multiple intelligent agent system via a localized generalization error approach', *International Conference on Service Systems and Service Management*, Chengdu, China, pp. 1-6.

Yeung, D.S., Ng, W.W.Y., Chan, A.P.F., Chan, P.P.K., Firth, M. & Tsang, E.C.C. 2007b, 'A multiple intelligent agent system for credit risk prediction via an optimization of localized generalization error with diversity', *Journal of Systems Science and Systems Engineering*, vol. 16, no. 2, pp. 166-180.

Yin, X., Han, J., Yang, J. & Yu, P.S. 2006, 'Efficient classification across multiple database relations: A CrossMine approach', *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 770-783.

Zadrozny, B. 2004, 'Learning and evaluating classifiers under sample selection bias', *21st International Conferenve on Machine Learning*, Bnaff, Canada, pp. 114-121.

Zadrozny, B. & Elkan, C. 2001, 'Learning and making decisions when costs and probabilities are both unknown', *Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, pp. 204-213.

Zar, J.H. 1999, *Biostatistical Analysis*, Prentice Hall, Englewood Cliffs.

Zhang, J. & Morris, A.J. 1999, 'Recurrent neuro-fuzzy networks for nonlinear process modeling', *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 313-326.

Zhou, J.G. & Tian, J.M. 2007, 'Predicting corporate financial distress based on rough sets and wavelet support vector machine', *International Conference on Wavelet Analysis and Pattern Recognition*, Beijing, China, pp. 602-607.

Zhou, R.W. & Quek, C. 1996, 'POPFNN: A pseudo outer-product based fuzzy neural network', *Neural Networks*, vol. 9, no. 9, pp. 1569-1581.

Zhu, X. 2005, *Semi-Supervised Learning Literature Survey*, Computer Science, University of Wisconsin-Madiso,, Madison, WI.

APPENDIX: ABBREVIATIONS

ANFIS	Adaptive Neuro Fuzzy Inference System
EM	Expectation Maximization
2SBR	Two-Step Bridged Refinement
2SFBR	Two-Step Fuzzy Bridge Refinement
AHP	Analytic Hierarchy Process
ASM	Alternating Structural Minimization
AUC	Area Under a ROC Curve
BAA	Broad Agency Announcement
BPNN	Back Propagation Neural Network
CBR	Case-Based Reasoning
CCA	Canonical Correlation Analysis
CMAC	Cerebellar Model Articulation Controller
CNN	Condensed Nearest Neighbour Rule
CRI	Compositional Rule Of Inference
DARPA	Defense Advanced Research Projects Agency
DIC	Discrete Incremental Clustering
DIFS	Domain-Independent Feature Space
Dm	Dissimilarity Function
DSFS	Domain-Specific Feature Space
DSFSs	Source Domain-Specific Feature Space
DSFSt	Target-Domain Specific Feature Space
DT	Decision Tree
EMV	Expected Membership Value
ENN	Wilson's Edited Nearest Neighbour Rule
FACDA	Feature Alignment-Based Cross Domain

FBR	Fuzzy Bridged Refinement
FBRDA	Fuzzy Bridged Refinement Domain Adaptation
FCBR	Fuzzy Case-Based Reasoning
FCC	Fuzzy Correlation Coefficient
FEWS	Financial Early Warning Systems
FF_NN	Feed Forward Neural Network
FFIEC	Federal Financial Institutions Examination Council
FGFW	Fuzzy Genetic Feature Weighting
FMCDM	Fuzzy Multi Criteria Decision Making
FN	False Negative
FNN	Fuzzy Neural Network
FP	False Positive
FRBC	Fuzzy Rule Based Classifier
FSFA	Fuzzy Spectral Feature Alignment
GA	Genetic Algorithm
GenSoFNN	Generic Self-Organizing Fuzzy Neural Network
GM	G-Mean
IEWS	Integrated Early Warning System
IMF	International Monetary Fund
IPTO	Information Processing Technology Office
KIS	Korea Investors Service
KLIEP	Kullback-Leibler Importance Estimation Procedure
K-NN	K-Nearest Neighbor
LSA	Latent Semantic Analysis
LVQ	Learning Vector Quantization
MAC	Manifold Alignment Using Correspondences
MAL	Manifold Alignment Using Labels
MDA	Multivariate Discriminate Analysis
MLP	Multilayer Perceptron

MMD	Maximum Mean Discrepancy
MSBR	Multi-Step Bridged Refinement Algorithm
MSFBR	Multi-Step Fuzzy Bridge Refinement
NB	Naïve Bayes
NLP	Natural Language Processing
NN	Neural Networks
OSS	One-Sided Selection
PCA	Principal Component Analysis
PCNN	Principal Component Neural Network
RBFN	Radial Basis Function Neural Networks
RMV	Refined Membership Value
ROC	Receiver Operating Characteristics
RS	Rough Sets
SCL	Structural Correspondence Learning
SEC	Security And Exchange Commission
Sm	Similarity Function
SMOTE	Synthetic Minority Oversampling Technique
SOM-NN	Self-Organizing Feature Map Neural Network
STDFS	Selected Target-Domain Feature Space
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TSVM	Transductive Support Vector Machine
TVR	Truth Value Restriction
UFS	Unified Feature Space
UMV	Unrefined Membership Value