Hierarchical clustering using K-Iterations Fast Learning Artificial Neural Networks (KFLANN)

Wong, Lai Ping

2007

Wong, L. P. (2007). Hierarchical clustering using K-Iterations Fast Learning Artificial Neural Networks (KFLANN). Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/2530>

<https://doi.org/10.32657/10356/2530>

Nanyang Technological University

Downloaded on 26 Apr 2023 23:16:13 SGT

NANYANG TECHNOLOGICAL UNIVERSITY

**HIERARCHICAL CLUSTERING USING K-ITERATIONS
FAST LEARNING ARTIFICIAL NEURAL NETWORKS
(KFLANN)**

Wong Lai Ping

**A thesis submitted to Nanyang Technological University in fulfillment of the
requirements for the Degree of Doctor of Philosophy**

Division of Computing Systems

School of Computer Engineering

2005

Acknowledgement

I would like to express my gratitude to all who granted me the opportunity to complete this thesis. I would like to thank Center of Computational Intelligence (C2I), School of Computer Engineering, Nanyang Technological University (NTU), Singapore for providing me the resources and the substantial support throughout my research works. It would have been difficult if not for the staff and postgraduate students from the C2I laboratory.

I am deeply thankful to my supervisor Dr Alex Tay Leng Phuan from NTU for his guidance, stimulation and encouragement that helped me throughout my research work. Also I would like to thank my co-supervisor Dr Xu Jian from A*Star Singapore Manufacturing of Technology for his valuable suggestions and thoughtful advice.

Also I would like to give special thanks my parents and friends who patiently provide the moral support during the journey through my research.

Table of Contents

ACKNOWLEDGEMENT	I
TABLE OF CONTENTS	II
LIST OF TABLES.....	V
LIST OF FIGURES.....	VI
ABBREVIATION.....	VIII
NOTATION	IX
ABSTRACT	X
1 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 MOTIVATION AND AIM OF THE RESEARCH.....	2
1.3 ORGANIZATION OF THE THESIS	8
2 LITERATURE SURVEY.....	10
2.1 UNSUPERVISED LEARNING	12
2.1.1 Clustering.....	12
2.1.2 Fast Learning Artificial Neural Network	13
2.1.3 KFLANN Algorithm Description	15
2.1.4 KFLANN Architecture.....	16
2.1.5 Important Observations of KFLANN.....	17
2.2 HIERARCHIAL ARTIFICIAL NEURAL NETWORK.....	18
2.2.1 Hierarchical Artificial Neural Network Models.....	19
2.2.2 Observations On Studied Hierarchical Models	20
2.3 HYBRID LEARNING	22
2.3.1 Feedforward Multilayer Perceptron (MLP) Network.....	23
2.3.2 Backpropagation Delta Learning.....	24
2.3.3 Hybrid Learning Models	30
2.3.4 Observations of the Studied Hybrid Learning Models	31
2.4 CURSE OF DIMENSIONALITY	32
2.4.1 Feature Selection	32
2.4.2 Feature Transformation (Compression)	33
3 KFLANN NETWORK	35
3.1 KFLANN NETWORK PARAMETER.....	35
3.1.1 Vigilance	35
3.1.2 Tolerance.....	38
3.2 DATA PATTERN RESHUFFLING.....	39
3.2.1 Experiment results on Data Pattern Reshuffling	44
3.2.2 Comparative Study of KFLANN with Other Clustering Methods.....	46
3.3 TOLERANCE INITIALIZATION USING THE STANDARD DEVIATION....	50
3.3.1 Normal Distribution Cases	51
3.3.2 Non-Normal Distribution Cases	52
3.4 MAHALANOBIS DISTANCE BASED KFLANN	53
3.4.1 Comparison between the Mahalanobis and Euclidean Distance Measurement.....	55

3.4.2	Results on Mahalanobis Based KFLANN Experiments	58
3.5	CHAPTER SUMMARY	60
4	HIERARCHICAL FAST LEARNING ARTIFICIAL NEURAL NETWORK.....	62
4.1	INTRODUCTION	62
4.2	HIEFLANN ARCHITECTURE	63
4.3	HIEFLANN ALGORITHM	64
4.3.1	Homogenous Feature Subspace Generation, a First Stage Clustering.....	65
4.3.2	Local Distribution Analysis, a Second Stage Clustering.....	67
4.3.3	Feature Compression.....	67
4.3.4	Final Convergence of Information, a Third Stage Clustering	70
4.4	CANONICAL COVARIANCE FEATURE COMPRESSION IN HIEFLANN ..	70
4.4.1	Rationale of Proposing C2FeCom.....	72
4.5	PERFORMANCE ANALYSIS OF THE HIEFLANN	73
4.5.1	Experimental Result And Discussion On The UCI Datasets.....	74
4.5.2	C2FeCom Revisited - HieFLANN Clustering Using the Wine Dataset.....	77
4.5.2.1	First Stage of Clustering, HFS Generation.....	78
4.5.2.2	Second Stage Clustering, Local Distribution Analysis	79
4.5.2.3	Third Stage, Feature Compression.....	80
4.5.2.4	Final Stage, Fusion of Information.....	81
4.6	DISCUSSION ON THE HIEFLANN.....	81
4.7	CHAPTER SUMMARY	83
5	HIERARCHICAL HYBRID LEARNING MODEL K-ITERATIONS FAST LEARNING ARTIFICIAL NEURAL NETWORK AND BACKPROPAGATION.....	85
5.1	THE NEED OF HIERARCHICAL HYBRID LEARNING MODEL	85
5.2	ALGORITHM AND ARCHITECTURE OF HIEFLANN_BP.....	86
5.2.1	Formation Of Homogenous Feature Space	86
5.2.2	Hybrid Learning.....	87
5.2.2.1	Performance Analysis Of Hybrid Learning Model KFLANN-BP	91
5.2.2.2	Hybrid Learning Model KFLANN-BP On Fisher Iris Flower Data Set.....	93
5.2.2.3	Comparisons Between the KFLANN_BP And KFLANN	94
5.2.3	Generalization Learning.....	95
5.3	PERFORMANCE ANALYSIS OF THE HIEFLANN_ BP.....	96
5.3.1	Experimental Analysis Of HieFLANN_BP On UCI Datasets	96
5.3.2	HieFLANN_BP On The Wisconsin Breast Cancer Data Set	97
5.3.2.1	Formation Of Homogenous Feature Space	97
5.3.2.2	Hybrid Learning.....	98
5.3.2.3	Generalization Learning.....	99
5.3.3	Comparative Study of HieFLANN_BP with other Hierarchical Model.....	100
5.3.3.1	Network Construction	100
5.3.3.2	Network Parameter.....	101
5.3.3.3	Learning Paradigm	102
5.3.4	Comparison Between HieFLANN_BP And HieFLANN.....	102
5.3.5	Comparison Between HieFLANN_BP With Non Hierarchical Network KFLANN And MLP-BP	104
5.4	CHAPTER SUMMARY	105
6	FEATURE SELECTION.....	108
6.1	IMPORTANCE OF FEATURE SELECTION	108
6.2	FEATURE PURITY	112
6.3	FEATURE RELEVANCE	113
6.3.1	Feature Relevance for Continuous Feature Types.....	113
6.3.2	Feature Relevancy for Nominal Feature Types.....	114

6.4	THE PROPOSED FEATURE EVALUATION INDEX: PURITY AND RELEVANCY.....	115
6.4.1	Methodology for PURE Feature Selection.....	115
6.4.2	Comparative Study of PURE Feature Selection With Other Feature Selection Methods.....	117
6.5	PERFORMANCE ANALYSIS OF PURE FEATURE SELECTION METHOD.....	120
6.5.1	Experimental Setup	120
6.5.2	Experimental Results of PURE Algorithm on UCI Datasets	122
6.5.3	Experimental Results of PURE Algorithm on Micorarray Datasets	126
6.5.4	Experimental Results Of PURE Algorithm On NIPS 2003 Datasets	127
6.6	INVESTIGATION OF HIEFLANN ON HIGH DIMENSIONALITY DATASETS	128
6.7	INVESTIGATION OF HIEFLANN_BP ON HIGH DIMENSIONALITY DATASETS	130
6.8	CHAPTER SUMMARY	132
7	CONCLUSION AND DISCUSSION.....	135
7.1	CONTRIBUTIONS	135
7.2	FUTURE DIRECTIONS	140
	REFERENCE	142
	APPENDIX A – KFLANN CONVERGENCE.....	156
A.1	CASE (A) WELL SEPARABLE CLASSES WITH EQUAL STANDARD DEVIATION WITH DISTANCE BETWEEN NEAREST CLASSES ARE GREATER THAN σ_0	158
A.2	CASE (B) WELL SEPARABLE CLASSES WITH UNEQUAL STANDARD DEVIATION WITH DISTANCE BETWEEN NEAREST CLASSES ARE GREATER THAN σ_0	160
A.3	CASE (C) WELL SEPARABLE CLASSES WITH EQUAL STANDARD DEVIATION WITH DISTANCES BETWEEN NEAREST CLASSES ARE SMALLER THAN σ_0	163
A.4	CASE (D) WELL SEPARABLE CLASSES WITH UNEQUAL STANDARD DEVIATION WITH DISTANCES BETWEEN NEAREST CLASSES ARE SMALLER THAN σ_0	165
	APPENDIX B – EVALUATION METHOD.....	170
B.1	RECEIVER OPERATING CHARACTERISTICS	170
B.2	NORMALIZED ERROR RATIO.....	170
B.3	NUMBER OF HIDDEN NEURON	171
C.1	UCI DATASET DESCRIPTION	172
C.2	MICROARRAY DATASETS	177
C.3	NEURAL INFORMATION PROCESSING SYSTEM (NIPS 2003) DATASETS	178
C.4	ARTIFICIAL DATASET.....	179
	APPENDIX D – PUBLICATIONS	180

List of Tables

TABLE 3-1 ACCURACY FOR ORIGINAL KFLANN AND MODIFIED KFLANN	44
TABLE 3-2 FEATURE COMPARISON BETWEEN	47
TABLE 3-3 COMPARISON ON ACCURACY BETWEEN KFLANN, LVQ AND SOM	48
TABLE 3-4 COMPARISON ON CLUSTERING ACCURACY FOR THE MAHALANOBIS DISTANCE METHOD AND EUCLIDEAN DISTANCE METHOD ON ARTIFICIAL DATASETS.....	59
TABLE 4-1 CLASSIFICATION ACCURACY FOR LOW DIMENSIONALITY DATASET	74
TABLE 4-2 CLASSIFICATION ACCURACY FOR MEDIUM DIMENSIONALITY DATASET	74
TABLE 4-3 CLASSIFICATION ACCURACY FOR HIGH DIMENSIONALITY DATASET.....	75
TABLE 4-4 COMPARISON ON THE ACCURACY OBTAINED FROM HIEFLANN, RBF AND C4.5	76
TABLE 4-5 HFSS FOR WINE DATASET	78
TABLE 4-6 FEATURE TOLERANCE FOR WINE 'S FEATURE SIMILARITY DATA	79
TABLE 5-1 EXPERIMENTAL RESULTS ON HYBRID LEARNING MODEL	91
TABLE 5-2 FEATURE TOLERANCE FOR IRIS DATA	93
TABLE 5-3 ACTUAL CLASSES CLUSTER MAPPING FOR IRIS DATA.....	93
TABLE 5-4 FEATURE COMPARISON BETWEEN KFLANN-BP AND BP.....	94
TABLE 5-5 ACCURACY OF HIEFLANN_BP FOR UCI DATASETS	96
TABLE 5-6 HFSS FOR WBC DATASET	98
TABLE 5-7 COMPARISON ON ACCURACY OF HIEFLANN_BP WITH HIERARCHICAL MODEL HIEFLANN	103
TABLE 5-8 ACCURACY COMPARISON OF HIEFLANN_BP WITH NON HIERARCHICAL MODEL.....	104
TABLE 6-1 COMPARISON BETWEEN LDA, PCA AND PURE	118
TABLE 6-2 ACCURACY FOR CONTINUOUS FEATURE TYPE UCI DATASETS USING PURE FEATURE SELECTION METHOD	122
TABLE 6-3 ACCURACY FOR CONTINUOUS FEATURE TYPE DATASET USING CORRELATION FEATURE SELECTION (CFS), CONSISTENCY AND RELIEFF FEATURE SELECTION METHOD.....	123
TABLE 6-4 ACCURACY FOR DISCRETE FEATURE TYPE OF UCI DATASET USING PURE FEATURE SELECTION METHOD	124
TABLE 6-5 ACCURACY FOR DISCRETE FEATURE TYPE DATASET USING CFS, CONSISTENCY AND RELIEFF FEATURE SELECTION METHOD	125
TABLE 6-6 DIFFERENCE IN ACCURACY FOR DISCRETE FEATURE TYPE DATASET USING PURE, CORRELATION FEATURE SELECTION (CFS), CONSISTENCY AND RELIEFF FEATURE SELECTION METHOD.....	125
TABLE 6-7 ACCURACY USING PURE FEATURE SELECTION METHOD FOR MICROARRAY DATASETS	126
TABLE 6-8 ACCURACY USING PURE AND CORRELATION FEATURE SELECTION (CFS) METHOD FOR NIPS 2003 DATASETS.....	127
TABLE 6-9 DIFFERENCE IN ACCURACY OBTAINED USING PURE AND CORRELATION FEATURE SELECTION (CFS) METHOD FOR NIPS 2003 DATASETS.....	127
TABLE 6-10 ACCURACY OF HIEFLANN FOR UCI DATASETS IN THE ORIGINAL FEATURE SPACE AND REDUCED FEATURE SPACE	128
TABLE 6-11 SF/D RATIO FOR MICROARRAY DATASETS	129
TABLE 6-12 ACCURACY OF THE HIEFLANN COMPARED TO C4.5 WITH PURE FEATURE SELECTION	130
TABLE 6-13 ACCURACY OF HIEFLANN_BP IN THE ORIGINAL FEATURE SPACE COMPARED TO C4.5 IN THE REDUCED FEATURE SPACE USING PURE FOR UCI DATASETS	130
TABLE 6-14 ACCURACY OBTAINED USING HIEFLANN_BP WITH PURE FEATURE SELECTION FOR UCI DATASETS.....	131
TABLE 6-15 ACCURACY OF HIEFLANN_BP COMPARED TO C4.5 (BOTH WITH PURE FEATURE SELECTION) FOR MICROARRAY DATASETS.....	132

List of Figures

FIGURE 2.1 KFLANN ALGORITHM.....	15
FIGURE 2.2 KFLANN ARCHITECTURE	16
FIGURE 2.3 MLP ARCHITECTURE	23
FIGURE 2.4 (A) MISO AND (B) MIMO MLP NETWORK	23
FIGURE 2.5 SIGMOID ACTIVATION FUNCTION	24
FIGURE 2.6 BACKPROPAGATION ALGORITHM	26
FIGURE 3.1 NUMBER OF CLUSTERS VERSUS VIGILANCE ON DIFFERENT UCI DATASETS	37
FIGURE 3.2 NUMBER OF CLUSTER VS TOLERANCE SCALING FACTOR ON THE STANDARD DEVIATION FOR DIFFERENT UCI DATASETS	39
FIGURE 3.3 ALGORITHM OF KFLANN WITH DATA PATTERN RESHUFFLING	40
FIGURE 3.4 2D DATA EXEMPLAR	41
FIGURE 3.5 FIRST ITERATION OF THE KFLANN CLUSTERING ON THE 2D EXEMPLAR DATASET	41
FIGURE 3.6 SECOND ITERATION AND THIRD ITERATION OF THE KFLANN CLUSTERING ON THE 2D EXEMPLAR DATASET	42
FIGURE 3.7 FINAL ITERATION OF THE KFLANN CLUSTERING ON THE 2D EXEMPLAR DATASET.....	42
FIGURE 3.8 DIFFERENCE BETWEEN ACCURACY OBTAINED USING THE KFLANN WITH DATA PATTERN RESHUFFLING AND THE ORIGINAL KFLANN	45
FIGURE 3.9 ACCURACY OBTAINED USING ORIGINAL KFLANN AND MODIFIED KFLANN ON DIFFERENT DATA PATTERN SEQUENCE FOR DIFFERENT UCI DATASETS	46
FIGURE 3.10 NORMAL DISTRIBUTION FUNCTION CURVE.....	51
FIGURE 3.11 CLUSTER CREATED USING MAHALANOBIS DISTANCE METHOD (DASH LINE) AND THE EUCLIDEAN DISTANCE METHOD (SOLID LINE)	55
FIGURE 3.12 DATA DISTRIBUTION WITH DATA POINT A AND B LOCATE FROM CENTROID OF ECLIPSE CLUSTER, Z.....	56
FIGURE 3.13 (A) DECISION BOUNDARY CREATED USING EUCLIDEAN DISTANCE METHOD AND (B) MAHALANOBIS DISTANCE METHOD.....	57
FIGURE 3.14 DIFFERENCE BETWEEN ACCURACY USING THE MAHALANOBIS DISTANCE METHOD AND EUCLIDEAN DISTANCE METHOD ON UCI DATASETS	58
FIGURE 3.15 (A) DIFFERENCE BETWEEN ACCURACY USING THE MAHALANOBIS DISTANCE METHOD AND EUCLIDEAN DISTANCE METHOD ON ARTIFICIAL DATASETS AND (B) DIVIDED BY STANDARD DEVIATION	60
FIGURE 4.1 HieFLANN ARCHITECTURE	64
FIGURE 4.2 HieFLANN PSEUDOCODE.....	64
FIGURE 4.3 PSEUDOCODE FOR HFS GENERATION.....	65
FIGURE 4.4 CONSTRUCTION OF HOMOGENOUS FEATURE SUBSPACE	66
FIGURE 4.5 EXAMPLE FOR CCM CONSTRUCTION	68
FIGURE 4.6 PICTORIAL VIEW ON C2FeCom PROCESS	70
FIGURE 4.7 PREPARATION PROCESS FOR C2FeCom	71
FIGURE 4.8 ACCURACY COMPARISON BETWEEN HieFLANN AND KFLANN ON UCI DATASETS	75
FIGURE 4.9 DIFFERENCE IN ACCURACY BETWEEN HieFLANN AND KFLANN IN THE STANDARD DEVIATION UNIT	75
FIGURE 4.10 COMPARISON OF THE ACCURACY DIFFERENCE BETWEEN HieFLANN, RBF AND C4.5.....	77
FIGURE 4.11 HieFLANN STRUCTURE FOR WINE DATASET	78
FIGURE 4.12 FEATURE SIMILARITY.....	79
FIGURE 4.13 6 KFLANN SUBNETS WITH THE RESPECTIVE OUTPUT FORM THE R_{yy}	80
FIGURE 4.14 CORRELATION COEFFICIENT ON THE FEATURE SPACE FORM THE R_{xx}	80
FIGURE 4.15 CORRELATION COEFFICIENT ON THE FEATURE SPACE AND KFLANN SUBNETS CLUSTERING OUTPUT FORM THE R_{xy}	81
FIGURE 5.1 PICTORIAL VIEW OF THE HieFLANN_BP NETWORK	86
FIGURE 5.2 PICTORIAL VIEW OF THE HYBRID LEARNING KFLANN AND MLP-BP	87
FIGURE 5.3 FLOWCHART OF THE HYBRID LEARNING MODEL OF KFLANN AND MLP-BP	89
FIGURE 5.4 STRUCTURE OF THE HYBRID LEARNING MODEL.....	90
FIGURE 5.5 (A) DIFFERENCE BETWEEN ACCURACY OBTAINED BY HYBRID KFLANN-BP AND MLP-BP (B) DIFFERENCE BETWEEN ACCURACY BY HYBRID KFLANN-BP AND KFLANN	92
FIGURE 5.6 GENERALIZATION LEARNING IN HieFLANN_BP WITH HYBRID LEARNING KFLANN AND MLP- BP NETWORK AT TOP MOST LEVEL	95
FIGURE 5.7 TRAINING AND TESTING ACCURACY FOR THE HieFLANN_BP MODEL.....	97
FIGURE 5.8 FEATURE SIMILARITY FOR WBC DATASET.....	98
FIGURE 5.9 HieFLANN_BP NETWORK STRUCTURE FOR WBC DATASET	98

FIGURE 5.10 PICTORIAL VIEW ON THE GENERALIZATION LEARNING OF HieFLANN_BP FOR WBC DATASET	99
FIGURE 5.11 THE HIERARCHICAL STRUCTURE OF THE NEOCOGNITRON NETWORK, S IS THE S-CELLS IN 2 DIMENSIONAL PLANE, C REPRESENTS C-CELLS LAYERING IN 2 DIMENSIONAL PLANE, V IS THE V-CELLS SITUATE IN V-PLANE. THE S-CELL AND C-CELL NEED TO HAVE NOTATION OF CELL TYPE, LAYER WHERE IT IS LOCATED, WHICH CELL PLANE IS IT AND THE LOCATION OF THE CELL IN THE PLANE.....	101
FIGURE 5.12 (A) DIFFERENCE IN ACCURACY OF HieFLANN_BP AND HieFLANN, (B) IN THE STANDARD DEVIATION UNIT	103
FIGURE 5.13 (A) DIFFERENCE IN ACCURACY OF HieFLANN_BP AND MLP-BP AND (B) IN THE STANDARD DEVIATION UNIT	104
FIGURE 5.14 (A) DIFFERENCE IN ACCURACY OF HieFLANN_BP AND KFLANN, (B) IN THE STANDARD DEVIATION UNIT	105
FIGURE 6.1 3D PLOT OF IRIS FEATURES	110
FIGURE 6.2 ALGORITHM FOR PURE FEATURE SELECTION	116
FIGURE 6.3 DIFFERENCE IN ACCURACY BETWEEN REDUCED FEATURE SPACE USING PURE FEATURE SELECTION METHOD AND THE ORIGINAL FEATURE SPACE FOR CONTINUOUS FEATURE TYPE UCI DATASETS	123
FIGURE 6.4 PLOT OF THE ACCURACY IN THE ORIGINAL FEATURE SPACE AND REDUCED FEATURE SPACE.....	132
FIGURE A.1 NEAREST DISTANCE BETWEEN NEAREST CLASSES AND OVERALL STANDARD DEVIATION FOR A SINGLE FEATURE DATASET OF 3 CLASSES	156
FIGURE A.2 DOT PLOT FOR CASE (A), NORMALLY DISTRIBUTED WITH EQUAL STANDARD DEVIATION FOR WELL SEPARABLE CLASSES, $\text{DIST}(C_1, C_2)$ IS GREATER THAN THE σ_o	157
FIGURE A.3 DOT PLOT FOR CASE (B) NORMALLY DISTRIBUTED WITH UNEQUAL STANDARD DEVIATION FOR WELL SEPARABLE CLASSES, $\text{DIST}(C_1, C_2)$ IS GREATER THAN THE σ_o	157
FIGURE A.4 DOT PLOT FOR CASE (C), NORMALLY DISTRIBUTED WITH EQUAL STANDARD DEVIATION FOR WELL SEPARABLE CLASSES, $\text{DIST}(C_i, C_j)$ ARE SMALLER THAN THE σ_o	157
FIGURE A.5 DOT PLOT FOR CASE (D), NORMALLY DISTRIBUTED WITH UNEQUAL STANDARD DEVIATION FOR WELL SEPARABLE CLASSES, $\text{DIST}(C_i, C_j)$ ARE SMALLER THAN THE σ_o	158
FIGURE A.6 DOT PLOT OF DATASET FOR CASE (A)	158
FIGURE A.7 KFLANN CLUSTERING OUTPUT AT FIRST ITERATION FOR CASE (A)	159
FIGURE A.8 KFLANN CLUSTERING OUTPUT AT SECOND ITERATION FOR CASE (A)	160
FIGURE A.9 DOT PLOT OF DATA DISTRIBUTION FOR CASE (B)	160
FIGURE A.10 KFLANN CLUSTERING OUTPUT AT FIRST ITERATION FOR CASE (B).....	161
FIGURE A.11 KFLANN CLUSTERING OUTPUT AT SECOND ITERATION FOR CASE (B)	162
FIGURE A.12 KFLANN CLUSTERING OUTPUT AT THIRD ITERATION FOR CASE (B)	162
FIGURE A.13 DOT PLOT OF DATASET FOR CASE (C).....	163
FIGURE A.14 KFLANN CLUSTERING OUTPUT AT FIRST ITERATION	164
FIGURE A.15 KFLANN CLUSTERING OUTPUT AT SECOND ITERATION	164
FIGURE A.16 KFLANN CLUSTERING OUTPUT AT THIRD ITERATION	165
FIGURE A.17 DOT PLOT FOR DATASET FOR CASE (D)	166
FIGURE A.18 KFLANN CLUSTERING OUTPUT AT FIRST ITERATION FOR CASE (D)	166
FIGURE A.19 KFLANN CLUSTERING OUTPUT AT SECOND ITERATION FOR CASE (D)	167
FIGURE A.20 DATA PATTERN SEQUENCE FOR CASE (D) AFTER SECOND ITERATION.....	168
FIGURE A.21 KFLANN CLUSTERING OUTPUT AT THIRD ITERATION FOR CASE (D)	168
FIGURE A.22 KFLANN CLUSTERING OUTPUT AT FOURTH ITERATION FOR CASE (D)	169

Abbreviation

Short Form	Definition
ANN	Artificial Neural Network
KFLANN	K-iterations Fast Learning Artificial Neural Network
MLP	Multi Layer Perceptron
BP	Backpropagation
MLP-BP	Multi Layer Perceptron with Backpropagation
HieFLANN	Hierarchical Fast Learning Artificial Neural Network
HieFLANN_BP	Hierarchical Fast Learning Artificial Neural Network-Backpropagation
PURE	Purity and Relevancy
MDS	Multidimensional scaling
PP	Projection Pursuit
CA	Canonical Analysis
CCA	Canonical Correlation Analysis
ICA	Independent Component Analysis
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
LVQ	Learning Vector Quantization
SOM	Self Organizing Map algorithm
SFOM	Self-organizing Feature Map algorithm
MSOM	Multiple Self-organizing Map
GHSOM	Growing Hierarchical Self Organized Map
HFS	Homogenous Feature Subspace
CCM	Canonical Correlation Matrix
ROC	Receiver Operating Characteristics
AUC	Area Under Curve
ART	Adaptive Resonance Theory
WTA	Winner Take All
DPS	Data Pattern Sequence
BB	Branch and bound
SFS	Sequential Forward Selection
SBS	Sequential Backward Selection
HFS	Homogenous Feature Subspace
Weka	Waikato Environment for Knowledge Analysis
MD	Mahalanobis Distance
ED	Euclidean Distance
NREM	Non Rapid Eye Movement
REM	Rapid Eye Movement

Notation

No.	Symbol	Definition
1	ρ	Vigilance
2	σ	Standard deviation
	σ_o	Overall Standard Deviation (OSD)
	σ_{C_i}	Standard deviation for i^{th} class
3	μ	Mean
4	δ	Tolerance
	δ_i	Tolerance for i^{th} Class
5	C_i	i^{th} class
6	$Dist(C_i, C_j)$	Distance between i^{th} class and j^{th} class
7	$Cov(X)$	Covariance of X
8	$CorrCoef(X)$	Correlation Coefficient of X
9	d	Number of features
10	n	Number of data patterns
11	S	Number of subnets
12	c	Number of clusters

Abstract

This thesis focuses on resolving the issues in high dimensional learning for neural networks. The curse of dimensionality has limited the growth of artificial neural network applications. While the curse is evident in current artificial systems, the stability and plasticity of the human biological brain shows no sign of this problem. Much of the information processing within the Prefrontal Cortex is performed as semantically transformed information.

The work represented here, is broken into 4 stages. At the basic strata, an attempt is made to define an original stable platform, The K-iterations Fast Learning Artificial Neural Network (KFLANN), by which hierarchical systems can be built. Effort was spent in defining such a platform that provides a consistent clustering behavior, regardless the Data Presentation Sequence (DPS).

The second stage of the work presents an original attempt to fuse the KFLANN with an approximation system by combining it with the Canonical Correlation Analysis (CCA). This is similar to the PCA, except that there is now a consideration of multiple dimensions. The resultant system is a hierarchical system which was called the HieFLANN.

The third stage presents a second model of hierarchical nature, but differs from the HieFLANN in concept. The resultant HieFLANN_BP model represents an attempt to cascade atomized network models to process segments of the original problem, only to recombine processed information at varying hierarchical layers.

A final stage of the work formalizes the preparation needed before data can become useful. An introduction of 2 parameters known as the Purity of the data and the RElevance of the data. With the two parameters, a PURE index was designed.

This work, through theory and experiments, could lead neural networks into solving larger problem domains that would otherwise have been marred by the curse of dimensionality.

1 INTRODUCTION

1.1 BACKGROUND

The human brain is a mysterious organ with many secrets still to be discovered and is persisting as a frontier for scientific challenge. An amazing property of human intelligence is its learning ability from arbitrary complex tasks and yet maintains a capacity to continue learning over the entire lifetime. This is referred to the plasticity and stability dilemma or sometimes called catastrophic forgetting. Neuroscientists and psychologists are still trying to understand this intriguing capacity so as to develop efficient learning frameworks to emulate natural learning. Engineers on the other hand are seeking biological inspired techniques that can enhance the practical engineering capabilities. This thesis focuses on developing models that mimic these behaviors that are suitable for clustering and learning high dimensional data. Much of the work existing in literature involves relatively small dimensional data sets.

Neural networks have contributed much to the field of classification and have been important in various real life applications such as data mining, speech recognition, handwriting recognition, vision analysis, bioinformatics and spatial data analysis. The core element of classification is feature analysis and the ability to differentiate characteristics within a given domain. To date there are varied approaches designed to address the feature analysis. These approaches encompass technologies that include neural network and statistical methods. Both segments possess merits that warrant attention of contemporary researchers. Artificial Neural Networks have the advantage over many methodologies in its brain-like adaptive learning. This adaptive nature enables it to respond effectively even under extreme environments.

Unsupervised learning usually refers to a form of clustering algorithm but it does not perform classification learning due to the absence of adaptive learning. There are however exceptions such as the Adaptive Resonance Theory model. There is a growing trend to hybridize both the supervised and unsupervised learning in order to provide the needed classification efficacy [93], [103].

One of the current difficulties in pattern classification process is with problems existing in high dimensionality feature spaces. When large or high dimensionality datasets are presented to typical classifier systems, longer processing times are expected. Neural networks have also been known to fail in convergence due to the sheer magnitude of feature attributes, also known as the curse of dimensionality. The performance of classifier systems tends to degrade when dealing with complex datasets. Furthermore, there is a tendency for large feature dimensionalities to contain features that do not contribute to the pattern recognition process. Thus, feature selection and feature compression are viable attempts to address the curse of dimensionality. Feature selection can be viewed as a form of supervision for selecting important features while compression can be deemed as transforming salient features within the original feature space, onto a reduced size feature space and still faithfully preserving the original contents.

1.2 MOTIVATION AND AIM OF THE RESEARCH

This thesis attempts to address high dimensional classification problems. This has been identified as a bottleneck because it retards the possible extension and proliferation of neural network research. It is also of interest as a model to understand how the brain may

extend information from the sensory percepts to the concept encapsulation stages evident at the higher cortical layers.

K-Iterations Fast Learning Artificial Neural Network (KFLANN)

Much of this work stems from the research in the Fast Learning Artificial Neural Network [4]. The motivation obtained from the work is elaborated in this thesis and the plausible links between the model and explanations of the psychological learning dynamics are presented. A new network known as the K-Iterations Fast Learning Artificial Neural Network was later derived from the original FLANN. The “ K ” within the implementation refers to the iterations needed in obtaining centroid stability. The KFLANN is an unsupervised clustering algorithm that uses a small number of epochs to complete its cluster analysis. In just the first pass, the network is able to form initial clusters. After each subsequent pass, the cluster centroids undergo rearrangement that eventually converge the data into consistent clusters. The KFLANN’s (or the FLANN’s) origins began with the hybridization of Kohonen Winner Take All [60] using a distance metric and the first introduction of the Adaptive Resonance Theory (ART 1) [40]. Similar to ART 1, the KFLANN uses a vigilance that is represented by ρ . It also uses a tolerance control variable δ to control localized variations within individual features. The KFLANN clustering process requires a network parameter initialization of both ρ and δ values. Based on these network constraint parameters, the clusters are produced and cluster membership assignments are made. The values of ρ and δ are determined heuristically from the characteristics of the domain. The implementation resembles the Leader Algorithm proposed in [43].

Hybrid Neural Model KFLANN and Multi layer Perceptron with Backpropagation (MLP-BP)

While the KFLANN addresses the issues in fast clustering and stability within the formation of classes, the continuous goal of this thesis focuses on higher dimensions. Referencing the human neural model, it is baffling how it seems to possess unlimited data storage of concepts that can be integrated continuously over a human's life time. Furthermore, the systematic formation of clusters is generally stable in the stereotype representations.

Leveraging from the KFLANN, we continue to investigate the possibility of unsupervised learning as an integral component for the learning process. The presence of an external 'teacher' can be useful for reinforced learning. It is not possible to have an infinite storage of KFLANN type neurons that encapsulate exemplar patterns. It is thus necessary to "off-load" stored exemplars onto a more efficient storage paradigm represented by well generalization mechanism. As a consequence of this suggestion, it was natural to extend the unsupervised learning KFLANN as the suitable candidate to quickly extract useful knowledge from the vast amount of data and then become the internalized "teacher" to extend training into some form of generalization learning system.

While it is possible to identify the KFLANN model as the means to form centroid stable clustering, the identification of networks that represent the long term portion of the memory is somewhat more difficult. The McCulloch-Pitts (M-P) type neurons proposed in 1943 [84] was responsible for many generations of neural network models. While it was able to mathematically provide a convergent behavior that mimics neural learning, the model by Hodgkin-Huxley in 1952 [49] was generally deemed by neuroscientists to be more precise and authentic in its representation of the actual neuron. This thesis does

not partake in this debate, but assumes in simplicity, that for the specific purpose of examining the hierarchical structure, the later stage of the long term memory system is sufficiently represented by the simple implementation using an Error Backpropagation type MLP network. The combination of both the KFLANN and MLP type network thus signifies the attempt to unify the learning paradigm where the short-term memory begins to train other network models found deeper in the brain that signify the long term memory. The alternative learning method is a hybridization of supervised and unsupervised schemes, evading the common path of creating a brand new learning algorithm for knowledge discovery. The discussion of the proposed hybridized supervised and unsupervised learning scheme provides experimental evidence that indicate better performance in general, over each of the single calibrated learning techniques. The result is an effective network that overcomes some of the limitations of each individual algorithm and produces benefits of a combined algorithm.

Hierarchical KFLANN (HieFLANN)

This thesis also investigates into the curse of dimensionality experienced by ANN models. It is interesting that human brain is able to grapple with extensive number of parameters without seemingly loosing focus. For ANN domain solutions, the curse of dimensionality has often crippled solutions when the input space is extensively large. Fukushima [36], [37] was able to conceptualize the need for hierarchical segmentation of the problem, but the research encountered difficulty in logically segmenting the information for hierarchical representation.

A hierarchical neural model consisting of multiple KFLANN subnets is developed in this thesis to evade the curse of dimensionality. This is presented as the Hierarchical Fast Learning Artificial Neural Network (HieFLANN). The HieFLANN implements the

divide and conquer strategy where the high dimensional data is divided into sub tasks of smaller segments that can be independently processed by a system of distributed sub-networks. The final solution is the fusion of solutions obtained from the sub-networks. Deep within the HieFLANN is a feature compression method based on Canonical Correlation Analysis (CCA). It serves as the engine for feature dimensionality reduction. As the original traits of KFLANN include unsupervised clustering, the HieFLANN extends this quality, allowing the automation of the decomposition task. This eludes the segmentation of the attributes for hierarchical representation.

Hierarchical Hybrid Neural Model HieFLANN_BP

A second hierarchical model, HieFLANN_BP, was derived from the HieFLANN. This motivation stemmed from the used of hybrid neural network models. While the HieFLANN provided a means of compression through CCA, the HieFLANN_BP hybrid model uses conventional network models to build the entire network structure.

Consistent with the motivations behind rehearsals, the hybrid sub-networks that reside within the HieFLANN_BP structure also perform a similar kind of ‘rehearsal’. Some positive aspects of this self-supervised hierarchical neural model include data driven initializing of network parameters, localized distribution analysis and parallelism.

Purity and Relevancy Feature Selection

Noisy data is a common occurrence and it becomes a nuisance as irrelevant features do not contribute positively to the knowledge discovery process. Removing irrelevant features proves well in facilitating the knowledge discovery process, reducing the computation time and increasing comprehensibility of final findings. This leads the discussions to two important feature evaluation indices called Purity and Relevancy.

Feature selection has been widely applied in knowledge discovery tasks to alleviate the negative impact of unnecessary features. To date, the feature selection approaches have been divided into two major groups namely the wrapper and the filter approaches. The wrapper approach relies on the adopted algorithm to select the features while the filter based method chooses features according to the defined feature evaluation index. The filter based method is preferable compared to the wrapper approach due to the complexity involve of the algorithm used. This becomes an impending issue when the feature dimensionality becomes large. The filter approach usually attaches with searching strategy to search for the most optimal feature subset. Different searching schemes may end at different feature subsets that yield inconsistent features selected. In addition, searching invokes some parameters that are difficult to set prior the searching process such as start points, end points and the search strategy.

Through the chapters of the thesis, several issues will be addressed in order to complete the objective mentioned in the beginning of this section. These issues include the following.

- To address the cluster consistency issue in unsupervised K-Iterations Fast Learning Artificial Neural Network (KFLANN) [116] clustering algorithm.
- To propose a novel adaptive hybrid learning strategy that uses a combination of the supervised learning Multi Layer Perceptron type network and unsupervised KFLANN.
- To propose novel hierarchical ANN model that evocates modular processing.
- To present successful feature compression and feature selection method for pattern recognition pre-processing step.

1.3 ORGANIZATION OF THE THESIS

This report consists of 7 chapters. Chapter 1 introduces the background and objective of this thesis. The motivation that inspired the research in using ANN to solve a knowledge discovery problem is described.

Chapter 2 contains the literature review section. This chapter gives a brief introduction to knowledge discovery. The historical development of ANN in the field of knowledge discovery is illustrated. The learning mechanisms adopted by ANN both supervised and unsupervised approaches are discussed. A summary on current issues encountered in knowledge discovery and some of the solutions proposed are provided. The supervised multilayer perceptron ANN with Backpropagation learning is also described. The unsupervised KFLANN is depicted at the end of the chapter.

Chapter 3 is an investigation into the dynamics of the vigilance ρ and the tolerance parameter. The data point reshuffling is introduced to address the cluster consistency. This was loosely tied to the rehearsal phase, either in consciousness or as REM processes during restful sleep [125]. Experimental testing and analysis on the KFLANN are given towards the end of the chapter.

This is followed by the proposed Hierarchical Fast Learning Artificial Neural Network (HieFLANN) architecture, and the learning algorithm is described in Chapter 4. Results analysis on the tested applications of HieFLANN on various real life problems are reported at the end of this chapter.

Chapter 5 dwells on the Hierarchical Fast Learning Artificial Neural Network interfaced with MLP type BP network (HieFLANN_BP). The rationale of developing the

hierarchical based hybrid learning model is discussed. Experimental evaluation and analysis are attached at the end of the chapter.

Chapter 6 describes the proposed feature selection method that makes use of the suggested feature evaluation index Purity and Relevancy (PURE). Feature purity measures how good a feature value distinguishes amongst classes. Feature Relevancy values the relationship between features and relationship with class content. Complementary benefits of purity and relevancy measurement provide an efficient feature goodness measurement for knowledge discovery.

Conclusions are outlined in Chapter 7. In this chapter, the contributions to knowledge are summarized.

2 LITERATURE SURVEY

The research in Artificial Neural Network (ANN) was inspired by the original biological neuron that provides practical paradigms for learning real and discrete valued vector functions from exemplars. These biological inferences eventually led to the development of the various connectionist models. As history records, the ANN research thrived from the 40's through to the 60's, but was severely affected in the 70's after failing to address linearly non-separable problems. It was not until after the proof of convergence in the Multilayer Perceptron (MLP) model that the field experienced a spontaneous revival in 80's. The connectionist model received a major boost of interest as researchers flocked into the area in somewhat similar fashion as the days of the gold rush. The Error Backpropagation (BP) learning algorithm was introduced by Werbos in 1974 [121], it was then popularized by Rumerhalt et. al. in 1986 [112] and gaining acceptance, many continued to develop the work in this area of ANNs. This flurry of excitement eventually evolved into a well-explored field with areas still undergoing changes. The amount of attention cast in this area has been overwhelming and has gained better acceptance in industrial applications [21], [28] and [104]. The chronological development of ANN follows.

William James (1909) [124]	Provided psychological background for ANN.
McCulloch and Pitts (1943) [84]	Modeled the single processing element.
Hebb (1949) [44]	Proposed Hebbian rule that stated "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently firing it, some growth process or metabolic changes take place in one or both cells".
Lashley (1950) [73]	Presented biological evidence for distributed processing.
John Von Neumann (1956) [57]	Imitated neuron function with telegraph relays and vacuum tube.
Rosenblatt (1957) [108], [109]	Proposed perceptron learning theory and proof the convergence of perceptron learning theory.
Widrow and Hoff (1960) [123]	Developed ADALINE (Adaptive Linear Neuron) architecture based on delta learning rule and the MSE criteria.

Minsky and Papert (1969) [87]	Commented that the single layer perceptron is unable to represent XOR problem that cause ANN receded during 1970s.
Grossberg (1970) [41]	Developed the theory on self-learning network embody with mathematical analysis.
Kohonen and Anderson (1972) [5]	Proposed content addressable memory concept and developed the linear associated memory neural networks.
Werbos (1974) [121]	Introduction of the Error Back Propagation learning algorithm.
Grossberg (1976) [40]	Investigated on the self-developed connections of neural network.
Kohonen (1982) [64]	Created the self organized network that adopted the competitive learning and winner take all concept known as self organizing feature map (SOFM).
Hopfield (1982) [50]	Created the Hopfield network with cost function based on the network state stabilization. Minimum energy level always achieved when the network converge.
Ackley (1985) [2]	Developed Boltzmann machine based on Hopfield network. Enhancement on Hopfield network is performed to compensate the false wells problem by using the stochastic activation function. The training paradigm implemented is known as simulated annealing with the temperature as the network parameter.
Rumelhart (1986) [112]	Discovered the Backpropagation learning algorithm led to refreshing of ANN. The multiplayer weight connections are updated based on the descent gradient delta learning.

The basic elements that differentiate ANN models are the network structure and learning scheme. The network structure includes the connection between neurons that can be described as feedforward, a feedback or a recurrent linkage. These linkages may play the role of excitation or inhibition. The learning scheme is categorized into supervised or unsupervised learning.

This segment of the thesis surveys both supervised and unsupervised. The purpose of this is to provide a platform by which both branches are later unified to model a biologically inspired learning paradigm using a hybrid of both the supervised and unsupervised models.

2.1 UNSUPERVISED LEARNING

The unsupervised learning is often taken as an equivalent to the clustering methodologies. This is due to the absence of domain knowledge and a lack of suitable supervising criteria during the grouping process. Clustering algorithms form clusters that are defined as a collection of data objects that are similar under predefined criteria. Members residing within the same cluster will possess similarities while members from other clusters will possess dissimilar traits. The quality of the cluster is determined by its ability to discover some or all the hidden attributes. The Fast Learning Artificial Neural Network (FLANN) [116] is an unsupervised clustering algorithm that will be extended upon. This will however be done in the light of other available clustering methods.

2.1.1 Clustering

Clustering has been widely applied in many disciplines to serve in data exploration applications or functions as a standalone tool to extract insightful information of data for other higher level algorithms.

As obtained from many applications such as in [27], [38], [53] and [118], the efficiency of a clustering algorithm was based on a set of standard measurements. Some of the measurement keys include the following [54], [55], [61], [111]:

Feature Type	It is necessity to handle any feature type of interval scaled (linear, continuous value), discrete (Boolean), categorical (nominal), ordinal or ratio value.
Cluster Shape	A flexible clustering algorithm should have no difficulty in representing cluster in any shape.
Cluster Size	It is preferable to generate clusters of non-uniform size that represent actual distribution of the data.
Domain Knowledge	Clustering algorithm with minimal interaction is preferable. The algorithm should facilitate the interruption of domain expert when it is desired.
Fault Tolerance	An efficient clustering algorithm should implicit with certain degree of fault tolerance in such a way that irrelevant attributes and frustrated environment should not cause severe effect on the clustering process.
Data Pattern Sequence	A good clustering algorithm is expected to be resilient sensitive to the data pattern sequence.

Clustering methodologies are too numerous, as such, only a relevant subset was reviewed here. While many of these algorithms have already been extended to improve on the clustering behaviors the review was done on the original algorithms.

Method	Positive Characteristics	Characteristic that need considerations
K-Means MacQueen (1967) [79]	<ul style="list-style-type: none"> • It is relatively scalable in processing large dataset • Relatively efficient $O(tkn)$, where n is number of patterns, k is number of clusters and t is the number of iterations 	<ul style="list-style-type: none"> • Sensitive to data pattern sequence. • Work well only when the mean of objects is defined. • Sensitive to noise and outliers. • Inapplicable to discover clusters with non-convex shapes or clusters very different size. • Heuristic method apply in determine the number of clusters. • It is difficult to interpret the clustering result.
SOM Kohonen (1988)	<ul style="list-style-type: none"> • Topological preservation. • Self organized network that is biological plausible. • Provide excellent approximation of the data set density. • Used to accelerate in a very large proportion the speed of convergence of classical simple competitive learning algorithm [29] • Enjoy the advantageous of competitive learning. 	<ul style="list-style-type: none"> • SOM does not minimize any define objective function; termination is not based on optimizing any model of the process or its pattern exemplar [117]. • Online SOM is order dependent and the final weight vectors depend on the data pattern sequence [117]. Prototype parameter estimates may be severely affected by noise or outliers. This is caused by the learning rate that computed as a function of the number of input presentation and node positions in the SOM grid. • The size of the output lattice, the step size and the size of the resonance neighborhood must be varied empirically from one data set to another to achieve useful result.
ART Carpenter & Grossberg (1987) [14]	<ul style="list-style-type: none"> • Solve the plasticity dilemma issue. • Enjoy the advantageous of competitive learning. • Emulated the attention and orientation system enable it acts as interconnection tool to other cognitive functioning. 	<ul style="list-style-type: none"> • It does not minimize any known objective function thus its termination is not based on optimizing any model of the process or its data [117]. • It implements the hard competitive approach that is more likely to be trapped in local minima. • Sensitive to data pattern sequence. • Sensitive to noise and outliers. • The learning rate is independent of time, thus it lacks of stability because of excessive plasticity.

2.1.2 Fast Learning Artificial Neural Network

FLANN was developed by Tay et al in [116] with the intent to use the Adaptive Resonance Theory (ART 1) as an analog based clustering neural network. This is

achieved by adapting the Winner Take All (WTA) property with the nearest neighborhood metric, commonly used in the Kohonen network. In reviewing literature, it was later discovered that the resultant FLANN algorithms resembles the Leader algorithm published in [43]. Further improvement was done to address numerical continuous value in FLANN II [30]. The FLANN II in its original form was plagued with sensitivity to the pattern sequence presentation of the data. Data presentation sequence sensitivity (DPSS) is seldom noticed in human behavior. As more general patterns of a class are presented, the human naturally classifies normal exemplars from odd ones. This happens even if the initial pattern exemplars presented were outliers. The elimination of the DPSS was proposed by incorporating the K-Means concept that resulted KFLANN [4]. The K-Means introduction improved the algorithm but some problems with the cluster stability persisted. Clusters were defined as stable if the clusters consistently reduce to a definite set of exemplars points regardless of the DPS.

The KFLANN was suitable for supervised and unsupervised training. As an unsupervised clustering algorithm it bore the following characteristics:

- It is a self-organized clustering network.
- No calibration data set was necessary.
- It can heuristically determine initial parameter settings through an analysis of the given data.
- It is able to deal with stability-plasticity dilemma by dynamically generating new categories from the training exemplars.
- It can be trained on-line.
- It can be trained within a single epoch.

2.1.3 KFLANN Algorithm Description

Similar to the Leader algorithm, the KFLANN algorithm is a short and compact algorithm. It uses simple principles and steps that can be easy to duplicate within small segment of code. The pseudocode is shown in Figure 2-1.

Figure 2.1 KFLANN algorithm

The initial step involves the network parameter setting for tolerance and vigilance values. The settings are domain dependent. The default setting for the Vigilance value usually ranges between 0.5 and 1.0. This specific value considers the permissible errors present in the entire feature during the clustering process. It is possible to have the vigilance setting less than 0.5 if there is information provided that a particular domain contains of very small portion (less than 50% for instance) of accurate feature values. When the vigilance setting is closer to unity, it implies that a more stringent clustering is adapted. Feature tolerance setting is the measurement of feature dispersion that is the radii of the cluster created.

The clustering process starts with the sequential presentation of data pattern. The cluster membership assignment is performed based on the tolerance and vigilance filtering as stipulated in step 3. Tolerance filtering is implemented base on the discriminating function $D[a]$ for all feature values. Tolerance filtering is executed by the decision hyperplanes encapsulated in the tolerance value defined. The vigilance filtering checks to ensure that a minimal number of features satisfy tolerance set.

In step 4, the KFLANN implements the WTA where the best match output is chosen from a pool of matched clusters to represent current data pattern. This process is facilitated by closest distance from data point to cluster centroid. The cluster membership assignment process in Step 5 is implemented using the WTA concept and the current data pattern is assigned to a single output node.

In the event where no similar clusters exist a new cluster will be formed. This is done in step 6 with the direct mapping of the weight connection using the data point vector.

2.1.4 KFLANN Architecture

Figure 2.2 KFLANN architecture

The KFLANN architecture is depicted in Figure 2.2 and it comprises of two layers namely the Input Layer (F1) and Output Layer (F2). Layer F1 is the direct map of the input patterns. The weight connection between input layer and output node is the input vectors. The Output layer is dynamically created to accommodate new exemplars from F1

that do not have representative patterns in the F2 layer. This architecture is simple to construct and is easy for parallelization purpose to duplicate.

2.1.5 Important Observations of KFLANN

The KFLANN algorithm requires an initialization of two network parameters. The KFLANN's network parameters are key to obtain an optimal clustering outcome. An in depth understanding of these two parameters will enable a better control of the algorithm and enhancements can be made efficiently.

As discussed previously, it is important to maintain consistent clustering results regardless the data pattern sequence. The K-Means concept was implemented to serve the cluster consistency purpose in the KFLANN [4]. KFLANN clustering was altered to compute cluster centroids upon presentation of any single data pattern. This implies that if the dataset contains n data patterns, there will be n centroid computations. This becomes too expensive especially for high dimensional data.

KFLANN clustering algorithm uses the typical Euclidean distance measurement in its data pattern similarity measurement. The Euclidean distance is a common approach adopted by many clustering methodologies. However the Euclidean distance has limitations when used in clustering. It creates spherical uniform clusters and by itself fails to account for possible relationships between features, weighing all features equally. Such formations may not be suitable for domains containing correlated features.

2.2 HIERARCHIAL ARTIFICIAL NEURAL NETWORK

As it is the focus of this thesis to investigate hierarchical structures, a survey was conducted on hierarchical networks. Hierarchical Artificial Neural Networks may consist of multiple neural networks cemented together in a self-organized form of components or constructed under the guidance from domain expert. Usually the networks resident within the hierarchical component can either be single neurons or complex neural architectures such as entire MLP networks or clustering models. Modularity and hierarchy are vital components in hierarchical ANN classifier systems. This is essentially used to subdivide a large or unstructured problem into smaller and manageable segments. Occam Razor's theorem [8], essentially stated that "Complexity should not be added unnecessarily". From the perspective of the subnets, hierarchical networks can provide the necessary simplifications.

Hierarchical implementations can simplify problems by reducing a large problem space into sub-problems that can be resolved locally. These localized qualities form the complete problem when fused together at the latter layers of the hierarchy. A face recognition system for example, can be built up from a set of localized systems resolving the position of the eyes, ears, mouth and nose of a person. While the fusion of these components may yield the recognition of a specific person, a localized network may also be responsible for determining if the person was smiling. This form of interaction is similar to the Katchalsky Sets (K-Sets) held faithfully by Walter Freeman and was actively researched by Kozma [71]. Certain portions can then be trained more specifically without affecting other areas that are out of the localized area. The entire architecture can be modified if the localized description changes, by training only levels involved in the modification. The hierarchy enables the network to tune itself towards recognizing a specific area with the most relevant knowledge.

This thesis advocates that modular network structures facilitate analysis. Ideally each sub-network is analyzed independently and a latter stage fusion is combined by a higher level neural network.

The hierarchical based ANN imposes a distributed implementation, possibility to improve fault tolerance. In modular implementations, each sub-network functions independently, affecting only the subnet. Thus global operations of the network are not affected severely.

The brain is divided into several compartments that serve different specification. Evidences are obtained from various sources [15], [48], [56] and [63], the brain is a distributed neural network in diverse regions that perform processing in a parallel fashion. The brain carries out the cognitive processing by splitting the task into smaller sub tasks [106] so that the original complex task is no longer difficult to handle. The cerebral cortex being the biggest part of the brain, it could not work properly without regional modularity [56]. Regional modularity is the division of the cognitive task to different hemispheres of the brain. In [47] and [88] it was shown that the splitting of a character during visual processing between different brain hemispheres improves visual word identification residing in a modular architecture.

2.2.1 Hierarchical Artificial Neural Network Models

Some early work by Fukushima using hierarchical structures, gave rise to the NEOCOGNITRON [36]. Another example is The Growing Hierarchical Self-Organized Map (GHSOM) [25] that constructed layers automatically. Each network was composed of an independent Self Organized Map (SOM) that adjusted its size according to the requirements of the input space.

NEOCOGNITRON is a massively parallel hierarchical MLP ANN founded by Fukushima in year 1979 primarily to handle the visual pattern recognition task. In each stage “salient” features are extracted to form a compressed representation from the output of the preceding stage. The setting up of the network involved three stages. These include the organization of the processing unit, interconnections between them and the functional description of the individual units. The modular units resident within Neocognitron represent features to be extracted, thus the designer has to initialize the network with foreknowledge of the problem space. As it turns out, this is a critical shortcoming of Neocognitron model.

GHSOM was developed by Dittenbach with the objective to represent the hierarchical structural of the underlying problem. It is totally unsupervised driven and a self-organized architecture without perquisite knowledge on the problem set. This is an attractive model for pattern recognition with its advantageous display including its adaptive distance measurement mechanism implemented. Unlike the conventional clustering methodology where it is too hard to determine the arbitrary size of clusters, this is not a problem for GHSOM. It inherits the merits of the upgraded SOM and can generate a nested sequence of partitions, determining the optimum number of clusters by enumerating the major clusters in an efficient and effective way. The most attractive feature of SOM is its topological preservation mapping that enables it to have the most reliable neighborhood relations’ transformation in the input data.

2.2.2 Observations On Studied Hierarchical Models

The Neocognitron relies heavily on the choice of parameters that determine how closely the unit input must correspond to the input it has been trained with, in order to elicit a

correct response. As such, the required human intervention is too expensive and creates a severe impact on the performance. As a consequence, it is rigid to develop simple yet efficient hierarchical ANN system.

Although the GHSOM seems to be fully automated in the network construction stage, the algorithm is dependent on the determination of the main parameters and how the weight vectors are initialized.

Investigations indicated that some unresolved issues of the hierarchical ANN exist. There is first the task of decomposition that minimizes the interference between sub-tasks resulting in maximal independence. Task decomposition is important for the modular network learning within the hierarchical network. This may be similar to finding a general method to determining the homogenous feature spaces (HFS) as proposed later in this thesis. The HFS is the feature space where features are more similar to each other as compared to the heterogeneous features. HFS is important for feature analysis since similar features can be encoded in compressed format. So far in the hierarchical model reviewed, no model addresses this issue.

Secondly, the distributed handling of the problem domain had not been considered by the hierarchical ANN models.

Thirdly, the curse of dimensionality has been seldom discussed, leaving as a side issue to be dealt with at a later stage. All hierarchical ANN models mentioned share the similar focus of handling large dimensionality dataset, but avoid looking into the curse of dimensionality issue. Most of the hierarchical ANNs aim to have autonomous network construction, however still require definitions of some network parameters. It is still

desirable to have a model that is self-organized and automatic in the network initialization.

2.3 HYBRID LEARNING

The two categories of learning schemes adopted by ANN are unsupervised and supervised. Each has its own merits and limitations. Supervised learning ANN may consist of individual cross linked neuron (processing unit) connected by weights like the multilayer perceptron (MLP) type network. Training takes place with the feeding of input data patterns together with associated target class to the network. The algorithm compares the network output with the actual target class to update weight connections. The iterative process repeats until the convergence criteria meet, where the goal is usually set to match the network output to the actual target as close as possible. In [10] and [82], it was mentioned that the generation of accurate actual target class can be very costly.

Classical supervised learning systems are capable of learning pattern sequences and inductively generalize the presented data patterns as stated in [32] and [130]. By learning the training set, these systems are then capable of generalization. Nigam [92] stated that the learning algorithms in this group require large amount of cleaned data exemplars in order to perform its learning and generalization well. It is known that cleaned data is difficult to obtain and this requirement may not be easily fulfilled. When the number of pattern exemplars increase, it is possible that a portion of exemplars contain inconsistent and conflicting characteristics. Such poor data conditions will surface as problems when MLP type models are eventually trained.

2.3.1 Feedforward Multilayer Perceptron (MLP) Network

Figure 2.3 shows the feedforward MLP network model. It consists of the input layer, hidden layers and output layers. The input layer connects the input signal to hidden layer through a series of weighted connections (W_{ki}). The neurons at hidden layer are again connected to the output layer through a second set of weighted connections (V_{mk}).

Figure 2.3 MLP architecture

The Error Backpropagation (BP) learning is a common learning algorithm in a multilayer perceptron model, denoted as MLP-BP. While there are variants in the MLP-BP network structure, this thesis investigates only the classical Multiple Input Single Output (MISO) and Multiple Input Multiple Output (MIMO) feedforward model.

Figure 2.4 (a) MISO and (b) MIMO MLP network

MLP-BP ANN model was inspired both by biological nervous systems and by mathematical theories of learning. For instance, BP algorithm is attached with mathematical proven learning theory. The model comprise of massively parallel interconnected neurons that work in the hierarchical way mimic the way biological nervous system do.

The representation of the MLP's output layer may be done in following formats:

- Binary Valued

The binary method uses dichotomous values of 0 and 1 target encoding for MLP-BP of type MIMO. If there are M classes, then there will be a M length of target string. For example there are 3 classes, the output value presented to the MLP-BP is [0 0 1] for the 1st class [0 1 0] for 2nd class and [1 0 0] for 3rd class.

- Continuous Valued

This Continuous method is for the MISO structure network. The target values are assigned within the range determined by the activation function used in MLP-BP.

Figure 2.5 Sigmoid activation function

For example, the value h in Figure 2.5 is the range bounded by the sigmoid activation function. The encoding will follow the formulation:

$$C_m = \frac{h}{M} * (m - 0.5) \quad (2-1)$$

Where C_m is the encoded value for m^{th} class to be used in the MLP-BP training. The unipolar sigmodal activation function has the range value of 1, the 3 classes will be represent in the value of 0.167, 0.499, 0.833. Those encoded values are used as training target value for MISO MLP typed network.

2.3.2 Backpropagation Delta Learning

The classical feedforward MLP-BP neural network adopted in this thesis comprises of the input layer, I , hidden layer (single hidden layer), H , and output layer O . This network acts

as a functional component to illustrate the possibility of hybridizing function approximation networks to the proposed KFLANN. Some preliminaries are explained here. Let $X = \{X_1, X_2, X_3, \dots, X_d\}$ be the data input space. d is the dimensionality of the feature space, number of input neurons is determined by d . The individual feature is denoted as $X_i = \{x_{1i}, x_{2i}, x_{3i}, \dots, x_{ni}\}$ represent i^{th} feature spans over n data samples, x_{ji} as input, $j \in \{1, 2, 3, \dots, n\}$ and $i \in \{1, 2, 3, \dots, d\}$. The input layer to hidden layer weight connection is given as $\mathcal{W} = \{W_1, W_2, W_3, \dots, W_s\}$, s is the number of hidden neurons $k \in \{1, 2, 3, \dots, S\}$, $W_k = \{w_{k1}, w_{k2}, w_{k3}, \dots, w_{kd}\}$ w_{ki} is the weight connection between i^{th} input neuron and k^{th} hidden neuron. The weight connection between hidden layer and output layer is $V = \{V_1, V_2, V_3, \dots, V_m\}$, O is the number of output neurons that determined by number of classes of the problem set and $m \in \{1, 2, 3, \dots, O\}$. $V_m = \{v_{m1}, v_{m2}, v_{m3}, \dots, v_{mk}\}$ and v_{mk} is the weight connection between m^{th} output neuron and k^{th} hidden neuron. α is the learning parameter and η is the momentum term introduce to favor the learning process. y_k^j is the activation (output) of k^{th} hidden neuron while o_m^j is the activation of m^{th} output neuron for j^{th} pattern sample. δ_m^j is the error signal produce by the m^{th} output neuron while δ_k^j is the error signal from k^{th} the hidden neuron. Algorithm of BP is given as follow:

*Step 1 Create MLP network by initiating input layer, hidden layer and output layer.
Randomly initial weights prototype.*

Step 2 While stopping criterion not met, perform the procedure

*For every input pattern, undergo
Feedforward phase*

Step 2.1

Present the input vector to the hidden layer

Step 2.2

Sum up the synaptic weights inputs signal for each hidden node

$$s_k = \sum_i^d w_{ki} x_{ji}$$

*Step 2.3**Compute the output signal using the defined activation function*

$$y_k = f(s_k)$$

*Step 2.4**Feed forward the hidden layer's synaptic output to output nodes**Sum up the weights inputs signal for each output node*

$$u_m = \sum_{k=1}^s v_{mk} y_k$$

*Step 2.5**Compute the output signal using the defined activation function*

$$o_m = f(u_m)$$

*Feedback phase**Step 2.6**Compute for each output node the error based target label attach (t_m^j) for the input pattern*

$$MSE = \frac{1}{2} \sum_{m=1}^O (t_m^j - o_m^j)^2$$

*Step 2.7**Compute output layer error signal*

$$\delta_m^j = (t_m^j - o_m^j) o_m^j (1 - o_m^j)$$

*Step 2.8**Calculate output layer weight correction*

$$\Delta v_{mk} = \alpha \delta_m^j y_k$$

*Step 2.9**Broadcast back the output layer error information to hidden layer**Compute the error signal at hidden layer*

$$\delta_k^j = y_k^j (1 - y_k^j) \sum_{m=1}^O \delta_m^j v_{mk}$$

*Step 2.10**Calculate hidden layer weight correction*

$$\Delta W_k = -\eta f'(S_k) X_i \sum_{m=1}^O \delta_m^j v_{mk}$$

*Weight update phase**Step 2.11**Output node*

$$V_m^{new} = V_m^{old} + \alpha \Delta V_m$$

Hidden node

$$W_k^{new} = W_k^{old} + \alpha \Delta W_k$$

*Test stopping criterion**End while*

Figure 2.6 Backpropagation algorithm

The characteristics of MLP-BP are summarized as follow:

Advantageous	
Gradient descent	BP learning paradigm can be seen as gradient descent based learning. Iteratively BP search through all possible solution space using gradient descent to reduce the error in the input output pairs mapping. It is able to converge to local minimum with respect to the network weights.
Feature invention	BP incorporates within the MLP model with hidden neurons layering at the intermediate layer provide the most intriguing properties such that new features are discovered that are not explicit in the input space. Detect the feature collinearities without intervention.
Arbitrary function approximator	BP take in input of any type include real value, discrete value or vector value during learning process. MLP-BP is able to approximate arbitrary function, The <i>Hecht-Nielsen Theorem</i> which is based on the Kolmogorov theorem to shows that any continuous function can be approximated by a feedforward network [45] and [46].
Functional dependency	Automatically take into consideration for any functional dependency without any instruction or network postulation.
Fault tolerant	With the distribution representation nature, it does not suffer severe from effect of outlier.
Simplicity	The learning method adopted is mathematical proven and easy to understand. Thus it is good for novice modeler provided more options of biological plausible simulation.
Input presentation	Handle straightforward the data without intermediate encoding. No assumption require on the data set. ANN takes in any type of attribute to learn the defined target function. The attribute value pairs may be highly correlated or independent of each other.
Limitation	
Long training time	It is a slow learner even adopts numerous heuristic speed up proposal such as learning rate adaptation, momentum setting, weight decay etc.
Black box	The learning process of weight connections incurred within the layering structure and interpretation of the mapping between input and output is difficult. Learned ANNs are less easily communicated to humans than learned rules.
Optimization of network structure	To date there is no optimal rule to determine the network structure including the number of neurons, connections paradigm, training iterations. Only heuristic approach is used to optimize the network structure given a problem set although there are vast research in the area of optimization of ANN network structure been performed.
Catastrophic interference	Hidden layer adaptations to new knowledge cause the interference effect.
Network connection	Only feedforward connection provide single direction on the information flow which is not biological plausible. Thus it does not really represent the real human cognitive functioning.
Interpretation of learned network	ANN is lack off the interpretation of the learned knowledge although it has amazing generalization ability.

While supervised MLP typed learning ANNs have been shown to handle generalization well, there are still issues that remain unresolved. While it is known that the initial network parameters do affect the final outcome, the MLP typed model can typically be only termed as a black box after the training is completed. This is sometimes not acceptable especially when there are a large variety of input attributes. There has always been a need to understand, in some way the interaction between input attributes and the final outputs. Even if there is a vague sense of plausible explanation, such explanations can suffice.

Furthermore, it is generally not possible to add incrementally to the training data during training phase. In the other words, online training is difficult to take place. If there is any change in data set, the learning algorithm must re-train with the entire training dataset. As a consequence this cause lengthy training time even small change of the training data set.

In contrast to supervised learning, unsupervised scheme like clustering only find clusters (groups) in data. However clustering is not considered a learning system unless it is able to adapt. Clustering is a simple process that assumes that the problem domain is complete and correct so that clusters generated represent the actual class distribution faithfully. However completeness and correctness are too stringent to most of the real world tasks, it is either impossible or too difficult to achieve.

Clustering systems provide a similarity match that attempts to link cause and effects between input attributes and output values. This provides a classification type of solution that often gains quick acceptance as plausible relationship tends to surface with the cluster formations. While these unsupervised classifiers provide meaningful relationship type clusters, they do not extend interpolative qualities that are sometimes needed in

applications. It is therefore desirable to obtain qualities of both types of methods and it is proposed that hybridization of the two networks be done to achieve the desired outcome.

The cognitive way of the human brain has been of interest since the time of the Egyptians. The brain has no problems performing classification tasks. While these tasks vary in complexity and utility of the different regions in the brain, there is an undeniable existence a hierarchical taxonomy such that different regions perform specific level of the process. Each region has been associated with specific functions and the entire structure exists as a system that performs modular execution. From the perspective of cognitive science and neuroscience there are different mechanisms taking place within the brain to bridge the large gap between for instance, visual stimulus and acoustic signal. The brain uses specialization of different structures such that its architecture is organized at various levels of neurons, networks, nodes, cortex area and their respective connections [19]. Further more different cognitive processes are heterogeneous. Those are evidence that multiple architectural representations and mechanism are involved in the brain during the information processing. Hence a hybrid neural model base mainly on artificial neural network is worthy to look into.

From the standpoint of the knowledge discovery, hybrid neural model have some advantageous as the results of the mutually complementary properties can be combined [114]. Unsupervised clustering has advantages of easy interpretation (cluster centroid, cluster density, cluster size and etc), simple encoding (for instance less network parameters) and explicit control. On the other hand, supervised learning like the multilayer perceptron with Backpropagation (MLP-BP) shows advantages for robustness, fault tolerance and learn adaptively. Those complementary merits inspire this thesis to develop a novel hybrid learning neural model.

2.3.3 Hybrid Learning Models

Hybrid model using the Learning Vector Quantization (LVQ) and Backpropagation (BP) learning algorithm was proposed by Frasconi et. al. in [34]. The authors claimed that the combination of LVQ and BP methods provided an effective and unique solution. Furthermore, they claimed that the LVQ system inherits from the BP learning, a classical statistical interpretation [74]. In [34], it was stated that BP learning with both sigmoidal and locally tuned Radial Basic Function (RBF) units imposes a well defined formulation for function optimization. The hybrid scheme introduced by Frasconi et. al. provides better formulation to LVQ for optimization purposes. At the BP side, it learns faster with the information provided by the LVQ system.

The Multiple Self Organized Map (MSOM) is a hybrid learning model proposed by Cervera [17]. It implements supervised and unsupervised learning using multiple self-organizing maps (SOM). The main idea is that classification is based on the SOM approximation of probability densities on the training data. With the known class probability density and adoption of the Bayesian Theory, SOMs tend to approximate the training data probability density in such a way to minimize the misclassification. The network training procedures are divided into two phase,

- The supervised phase
- The unsupervised phase

The supervised phase perform a separation stage where the training data is partitioned according to the class content and each SOM will take each disjoint subset of training data. The second phase uses the standard SOM unsupervised training on each SOM created from phase one. The author claims that this network is computationally economic and the network can be implemented in parallel without difficulty. The noticeable benefit

is the Divide-and-Conquer strategy that systematically reduces a complex problem into several manageable problems.

Another similar type of hybridization is the hybrid connectionist network that combine SOM and MLP networks [127]. The network performs automated document classification using a SOM to create appropriate clusters and the clusters are subsequently used for MLP training. The complementary benefits are clear, SOM encode compressed form features to facilitate MLP training while MLP contributes to the formulation of the function approximation.

2.3.4 Observations of the Studied Hybrid Learning Models

Hybrid model (LVQ and BP) developed by Frasconi performed well provided the LVQ system cluster the dataset accurately with the predefined cluster radius. The heuristic method is used to determine this radii parameters may cause the entire method degrade. Hybrid models that rely heavily on the network parameter should be avoided.

The MSOM method still relies on the probability density approximation that can be hindered by larger feature dimensionality. There were no considerations made to neighborhood relations between the individual SOMs which may actually contain vital information that is necessary for classification.

The hybrid model of SOM and MLP is suitable for moderate sized input feature dimensionality as it implements a single calibrated SOM and MLP. It is common knowledge that conventional MLP models suffer the curse of dimensionality and this limits the model to a certain size. The model was verified with a simple dataset and author mentioned that further refinement was needed.

2.4 CURSE OF DIMENSIONALITY

Knowledge discovery has been an intriguing research area and some critical issues remain unresolved. In this thesis, the curse of dimensionality issue will be taken into consideration. The handling of large dimension data is a heavy burden in many computational models and it degrades the classification accuracy. There is no neural network architecture that escapes the curse of dimensionality. It is just a degree of avoidance.

The most straightforward approach to address the large dimensionality feature is to reduce the dimensionality. Two commonly used approaches are feature extraction and feature selection. In both methods, there are supervised and unsupervised driven. Majority of feature selection methods are supervised driven and it can be categorized into wrapper and filter approach.

2.4.1 Feature Selection

Feature selection tends to evaluate the entire feature space and retrieve salient features. It performs an exhaustive search for m features from a d dimensional feature space (where $m \ll d$). A straightforward method is to scan all possible permutations of the features subset leading to $\binom{d}{m}$ possible combinations. The feature subset with the minimum classification error bearing the largest change is considered to be the salient feature. Cover et al commented that there was no guarantee of optimal feature subset using the exhaustive sequential feature selection procedure [20].

The Branch and bound (BB) method [91] searches the bounded enumeration of feature space. The optimality is guaranteed provided the criterion function satisfies the monotonicity constraints. The limitation of BB is the worst case of its exponential complexity.

Sequential Forward Selection (SFS) selects single features at a time and evaluates the criterion function [3]. If the inclusions degrade the performance then the added feature is dropped. The algorithm is computationally intensive since too many possible enumeration of the feature combinations exist. The Sequential Backward Selection (SBS) works in similar way but all features are presented at the initial stage and drop one by one to obtain the optimal subset.

2.4.2 Feature Transformation (Compression)

Feature compression is a process of feature space reduction from original feature space. The successfully reduced feature space should retain the information of the original feature space faithfully. Most of the existing methods stem from statistical methods, of which, some are discussed later.

Commonly used statistical methods include Principal Component Analysis (PCA) [58], Independent Component Analysis (ICA) [1], Canonical Analysis (CA) and Projection Pursuit (PP) [51]. These approaches determine an appropriate reduced dimensionality feature space either in linear or nonlinear way. ANN based feature extraction also underwent exhaustive investigation, there are also ANN based PCA, CA and PP.

PCA seeks the directions that maximize the spread of data irrespective of the group structure. It computes the m largest eigenvectors of $d \times d$ covariance matrix of the d

dimensionality input space. The infant PCA is linear transformation, the nonlinear based PCA has been proposed to deal with nonlinear feature space such as kernel PCA. This is the major limitation since it does not emphasize group structure, thus it does not really serve the purpose of classification.

ICA is more appropriate for the case where non Gaussian distribution since it does not depend on the second order property of data. It assumes that problem set are linear mixtures of some unknown latent features that are non-Gaussian and mutually independent, thus they are called the independent components of the observed data. This assumption is too strict in real life pattern recognition problem task where collinearity is always the case.

CA originally aimed to seek the correlation amongst sets of predictor variables (features) and criterion variables (group structure). The details are discussed in Chapter 6.

In PP, the data is projected linearly. The projection reveals the non-normally distributed structure of the data set. This is done by assigning a numerical "interestingness" index to each possible projection, followed by maximizing the index to seek the best projection. Interestingness index can be linear discriminant index. This method tries to express some nonlinearities thru the projection choosen. However if the data set is high-dimensional and highly nonlinear it may be difficult to find the linear projections.

The literature survey covered in this chapter summarizes some of the readings relevant to the eventual direction of this thesis. Many more articles were studied but were not included to prevent cluttering of irrelevant information.

3 KFLANN NETWORK

This chapter focuses on the dynamics of the K-iterations Fast Learning Artificial Neural Network (KFLANN). Since the KFLANN is the foundational component of the rest of this work, it is crucial that the dynamics of the algorithm is explained in detail. This portion provides a systematic evaluation where quantifiable experimental results are presented to explain why certain heuristics are suitable for use as initialization parameters. It provides a detailed description of the vigilance and tolerance parameters which characterize the network behavior.

3.1 KFLANN NETWORK PARAMETER

The two important network parameters in the KFLANN model are vigilance (ρ) and tolerance settings (δ). The vigilance factor concept was taken from the Adaptive Resonance Theory (ART) [14], [39], [40] where it parameterizes the attention aspect of the network. The tolerance setting was introduced to the ART type model such that it would also possess a similar behavior to the type of nearest neighborhood competition found in the Kohonen networks. These network parameters are crucial factors in determining the clustering resolutions that affect the classification performance directly.

3.1.1 Vigilance

The vigilance was a parameter that originated from [14]. It was designed as a means by which it could influence the degree of a long term memory trace match. The larger the vigilance value, the stricter the match. A small vigilance value would thus relax the matching criteria. The vigilance value in the KFLANN is similar and it is used to

determine the number of the attributes in the exemplar that are similar to the selected long term memory trace. Given a pattern consisting of 10 attributes and if only 8 matched attributes was needed for needed for consideration into the same cluster, then the vigilance should be 0.8. The vigilance formulation is given by:

$$\rho = \frac{f_{match}}{d} \quad (3-1)$$

Where f_{match} is the number of the features needed in order to classified as a same cluster and d is the total number of features.

It is usually a norm to have the vigilance set between 0.5 and 1.0, where 1.0 is the most stringent match and 0.5 requires only half the number of features to be within tolerance. The setting of the vigilance value is sometimes deemed as a trial-and-error process where a suitable value can be sought by systematically perturbing the value till the desired clustering behavior is achieved.

As a simplified illustration, the vigilance tuning may begin with a low value, of 0.5 and increase by small perturbations +0.1 for subsequent iterations. As the vigilance continues to undergo perturbations, the clustering accuracy is constantly used to verify the fidelity of increasing the vigilance. If the increment improves the clustering performance, the process is continued until the clustering accuracy begins to degrade or the maximum number of iterations has been achieved. Several systematic or heuristic schemes for making the small changes can be adopted. These can follow search strategies optimizing the necessary search function criteria and thus would not be discussed as part of this work. In experiments discussed in this thesis, it was assumed that preparatory filtering was performed prior to clustering.

While the different heuristic approaches can be used to set the vigilance value, the guidance from a domain expert is usually beneficial and helpful. The presence of a

domain expert is advisable for real world applications where there is an availability of known feature boundaries.

When the vigilance setting is high, cluster generations will tend to be larger. This is due to the more stringent requirements imposed on the features to be within the tolerance range. Figure 3.1 is the plot of the number of clusters (vertical axis) versus vigilance value (horizontal axis) of 9 individual UCI datasets (detail description of the datasets is discussed in Appendix C). By varying the value of the vigilance against constant feature tolerance setting, the number of clusters created was recorded for each tested dataset.

Figure 3.1 Number of clusters versus vigilance on different UCI datasets

The trend shows that for all tested datasets the number of clusters increases as the vigilance value is increased. Although the goodness of a particular clustering has no direct correlation with the number of clusters, this exercise illustrates that the more attentive the KFLANN is to details, the more classes are formed due to a stricter vigilance. This corresponds to the vigilance in the ART-type networks

3.1.2 Tolerance

While the vigilance value governs a global variation in the input features, the tolerance is a localized control, affecting only individual features. The tolerance setting is the measurement of a particular feature consistency that measures the maximum allowable fluctuating range of the specified feature. When domain experts are not available to recommend tolerance values, a reliable heuristic is recommended for deriving the tolerance values. This is known as the standard deviation method.

By definition, the standard deviation of a set of data is the measurement of dispersion of sampled variables from the mean. The tolerance computation using the standard deviation is given in equation:

$$\sigma_i = \sqrt{\frac{\sum_j^n (x_{ji} - \mu_i)^2}{n}} \quad (3-2)$$

Where x_{ji} is the i^{th} feature of the j^{th} data patterns and μ is the respective mean value

Just as the vigilance value can affect the cluster size, the individual tolerance values can also affect the cluster size.

Another evaluation was conducted to illustrate the relationship between the tolerance value and the number of clusters formed. Results indicate, the smaller the tolerance value, the greater the number of clusters. A plot of graph on the number of clusters versus a standard deviation range of 0.3 to 2.0, Similar to the previous cases in Figure 3.1, the same 9 UCI datasets were used.

Figure 3.2 Number of cluster vs tolerance scaling factor on the standard deviation for different UCI datasets

Figure 3.2 is the plot of the number of cluster versus the vigilance setting on 9 UCI datasets. The horizontal axis is the standard deviation scaling factor. A value of 0.5 implies that the standard deviation was multiplied by a factor of 0.5 and the tolerance value will become smaller. On the other hand, a scaling factor of 1.5 makes the tolerance value larger. The cluster size will thus become smaller as the tolerance value becomes smaller and vice versa.

3.2 DATA PATTERN RESHUFFLING

As mentioned in Chapter 1, the KFLANN was derived from the Leader type algorithm first published in [43]. While the algorithm is efficient in clustering, it encounters problems with cluster inconsistencies caused by varying Data Presentation Sequences (DPS). DPS is defined as the order by which data exemplars arrive to the network. The sensitivity to DPS was addressed by Tay, but the solution partially solved the problem [4]. The technique by Tay required that every pattern exemplar presentation perform the entire calculation of cluster centroids. This recalculation at every stage imposed a heavy

computational overhead, making it computationally intensive. This problem more evident in large data sets. Data patterns reshuffling was introduced to overcome this limitation and the results were published in IJCNN 2003 [126]. This work is also presented here as part of the thesis. The data points that were closest to the centroids were moved towards the top of the data list for the next iteration of KFLANN clustering. A modification on the KFLANN algorithm is provided below in Figure 3.3 and step 7 was added to improve the original algorithm presented in the survey found in Chapter 2.

Figure 3.3 Algorithm of KFLANN with data pattern reshuffling

Step 7 shuffles data patterns closest the cluster centroid points to the top of the new data list. The old cluster results are discarded and the KFLANN algorithm is executed again using the new data list. This process ensures that the previous cluster centroids are first to be clustered in the next epoch and these centroids are subsequently spread across the domain space before individual exemplar cases are presented. These latter presentations

then become members of the Voronoi spaces defined by the initial centroid seeds introduced at the beginning of the epoch. It performs a re-arrangement that stabilizes the membership space based on the tolerance values.

The example with 2 classes in Figure 3.4 is used to illustrate how the KFLANN reshuffles data patterns and how centroid stability is achieved. Assume that the data points initially presented start from P_1 , P_2 , P_3 and P_4 . The 2 classes are differentiated using circles and squares as shown in the figure. Without dwelling into the details of δ and ρ network parameters, assume the algorithm adopts the initial 4 data points as centroids during the first iteration.

Figure 3.4 2D data exemplar

The first iteration of the KFLANN clustering will produce 4 clusters as shown in the Figure 3.5 (a).

Figure 3.5 First iteration of the KFLANN clustering on the 2D exemplar dataset

After the first iteration, a re-computation of centroids leads to the relocation of the centroids to the new cluster means, indicated by the arrows in each clusters in Figure 3.5 (b).

Figure 3.6 Second iteration and third iteration of the KFLANN clustering on the 2D exemplar dataset

In the second iteration, as the consequence of relocating data points closest to the cluster means to the top of the data list, an apparent alteration of the clustering is seen. 4 clusters were created as shown in Figure 3.6 (a) with centroids A, B, C and D. Notice that 4 centroids have shifted from the original centroids in Figure 3.5. Centroids A and B tend to move closer together as compare to the first iteration and centroid C and D have similar movements. This is because the mean calculations have a central tendency, moving towards the position of highest concentration. Due to the instability of the centroids, the KFLANN clustering proceeds to the third iteration as shown in Figure 3.6 (b). At this iteration, 3 clusters were formed with centroid A, C and D. Centroid A has been stabilized while centroid C and D will continue to move in the direction shown in the arrow. At the fourth iteration (last iteration), the 3 clusters stabilize with centroids and clusters shown in Figure 3.7.

Figure 3.7 Final iteration of the KFLANN clustering on the 2D exemplar dataset

2 classes of 2D data exemplar above have been clustered using KFLANN resulting 3 clusters. The data pattern reshuffling process resident at each iteration led to clusters stabilization.

In the psychophysical analogy, the centroid of the cluster can be deemed as a stereotyping of the specific class and the other non-centroid members belong to the class stereotype. It is suggested that there is a possibility that the formation of this stereotyping in daily experiences is linked to a re-organization of recently learnt information and combined with already learnt information. We suggest that such information reorganization is necessary to support the plasticity and stability dilemma. This trend of thought was inspired by the plausible effect that when humans enter sleep, there is a reorganization of information learnt so that it can be organized for long term memory storage. Physiological studies into sleep patterns show that the Non Rapid Eye Movement and Rapid Eye movement stages seem to activate many neural activities. The interesting difference in some of these stages, as compared to waking periods, is the lack of external stimulus and the subject is still able to experience active interaction within the dreams. In fact, the Thalamus that is responsible for all sensory inputs to the brain is completely turned off, while the rest of the brain seems to replay experiential activities forward and backwards within the brain [77]. It is a plausible view, that these activities perform a re-arrangement of the learnt experiences and provides a session where information is committed to the longer term memory. This is however a view held without substantial proof that this work is not attempting to resolve. However the REM and NREM stages are known recorded evidence found in neuroscience and neuropsychological literature.

The KFLANN algorithm extends this psychophysical observation to the model where reshuffling is the necessary session that will iteratively stabilize the centroids (stereotypes).

This process will continue as long as the centroid stabilization is not achieved. Centroid stabilization is achieved when the data patterns to be reshuffled have already been on the centroid list previously. The algorithm is said to have converged when centroid stabilization is achieved. The reshuffling process involves the movement of those data points closest to the centroids to top of the data list imply the change on the data pattern sequence. The subsequence clustering process repeats on the re-shuffled data list. In all the experiments, it was found that a 5 iterations was typical for the centroid stabilization process. Further investigation as to why a typical iteration of 5 seemed to suffice was not pursued, as the focus of this work was to extend hierarchical modeling. Till this stage, it is sufficient to accept that 5 iterations of re-shuffling is sufficient to achieve acceptable convergence. This number was derived purely from experimental means without a convergence proof.

3.2.1 Experiment results on Data Pattern Reshuffling

Experiments were subsequently performed to compare the performance of the KFLANN with the original KFLANN algorithm without data pattern reshuffling. Table 3-1 compares the relative accuracies of the results obtained using the two different methods. There is an indication that an overall improvement and that the KFLANN generally seems to achieve better results.

Dataset	Original KFLANN (%)	Modified KFLANN (%)
Iris	96.00	100.0
Thyroid	82.00	92.00
Glass	54.28	69.05
Wisconsin Breast Cancer	82.83	91.77
Wine	89.28	90.77
Heart	76.67	77.14
Segment	70.87	88.43
Waveform	78.35	77.40
Ionosphere	70.15	71.15
Dermatology	92.65	94.44
Spambase	83.70	86.01
Sonar	67.50	67.54

Table 3-1 Accuracy for original KFLANN and modified KFLANN

A difference in the accuracies was graphed. Figure 1-8 shows the difference of the accuracy results of the original KFLANN from the modified KFLANN.

Figure 3.8 Difference between accuracy obtained using the KFLANN with data pattern reshuffling and the original KFLANN

In the selected datasets, the modified KFLANN performed better than the original one for most of the datasets. The experimental results suggested that data pattern reshuffling facilitates the algorithm to find centroids location at the high density area of the data distribution to achieve better clustering performance.

Another experiment was conducted to verify that the consistency was indeed achieved by the reshuffling. Each dataset was permuted randomly to obtain 5 sets in different ordered sequences. Both the original KFLANN and modified KFLANN were used to cluster the 5 datasets. Figure 3.9 plots the accuracy (vertical axis) obtained on the 5 sets (horizontal axis) differently sequenced datasets. The original KFLANN was observed to have a drastic fluctuation in the accuracies while the modified KFLANN maintained a relatively consistent accuracy throughout the 5 data sets. This indicated that the modified KFLANN was able to overcome the variation in the different data pattern sequences. On the other hand, the original KFLANN is not as consistent as the modified KFLANN. This evident fluctuation for different data pattern sequence implies the vulnerability of the algorithm to the data pattern sequence. From the previous experiment, it is also fair to

note that the accuracy in the modified KFLANN tends to be higher than of the original KFLANN.

Figure 3.9 Accuracy obtained using original KFLANN and modified KFLANN on different data pattern sequence for different UCI datasets

The experimental results support the hypothesis that data pattern reshuffling facilitates the KFLANN clustering in pursuing the cluster consistency. If indeed the centroids can be considered a means for stereotyping, there will be a more consistent stereotyping for further generalization in the hierarchical pursuits in the later chapters to follow.

3.2.2 Comparative Study of KFLANN with Other Clustering Methods

KFLANN is a neural network based clustering employs unsupervised learning scheme, thus clustering algorithms within these group are compared. Learning Vector Quantization (LVQ) [65], [66], Adaptive Resonance Theory (ART) [16] and Self organize map (SOM) [67], [69] are neural network based clustering model. Comparison of their characteristics with KFLANN clustering model were summarized in Table 3-2. The comparisons are done on the basic algorithm of each counterpart and omits the enhanced version. The primary stage of LVQ, SOM and ART was used for comparison in this thesis. It is not fare to compare a newly proposed algorithm with a well established

algorithm. Furthermore, there is a lot of expansion space for KFLANN that can be planned for future work.

Feature	KFLANN	LVQ	SOM	ART
Network Topology	2 layers	2 layers	2 layers	2 layers
Learning paradigm	Unsupervised	Supervised	Unsupervised	Unsupervised
Number of cluster	No pre-defined	Pre-defined of number of clusters	No pre-defined	No pre-defined
Iterative execution	Small number of iteration to converge (~ 5)	Iterative	Iterative	Iterative

Table 3-2 Feature comparison between

From network topology perspective, it was noticed that there are two layers in all four different clustering models discussed here. Except the LVQ model, input layer plays the role of input data integration. In LVQ, the first layer is a competitive non-linear neural network. Output layer of the KFLANN and ART consists of nodes that are created dynamically during the clustering process. LVQ's output layer is a layer of linear classifier. Second layer of the SOM model usually made up of nodes that are arranged in one (or two) dimensional array (also known as topological map). KFLANN and ART network models more attractive since network structure is created based on input data compared to LVQ and SOM.

When a pattern is presented to the ART network, it 'resonates' a few times between the layers before a response is issued from the output layer. In the KFLANN, centroids are shifted until convergence state achieved through the iterative process. When comparing the ART and KFLANN network in their learning methodology, they share the same mechanism of a pattern matching process. When an input pattern is presented, the matching process search thru the memory presentations, if the search ends at an established node, the node may incorporate new knowledge from the presented input or remain the same. If the search cannot find a match representation node, a new node is created to learn this new input pattern.

For LVQ network, the first phase of learning is to classify input features into similar sub classes (similar to SOM), second phase of learning transforms the competitive layer's classes into user labeled target classifications. LVQ and KFLANN both adopt competitive

learning. In LVQ, the competitive learning classify input vectors base on the distance between input vectors, it learns the distribution of inputs by dedicating more nodes to classifying parts of higher densities of input features. This is however different from the way WTA was applied in the KFLANN, it is a process of cluster membership assignment choose a output candidate where new member is introduced to.

Training in SOM network resembles vector quantization algorithms; it is distinct in addition to the best matching weight vector (the vector that will learns by adjusting it weights and its neighbourhood nodes are updated). Vector quantization learning in SOM does not exists in the KFLANN network, SOM similar with the KFLANN is that both select the best matching vector (a.k.a. output node) for learning of a given data pattern.

In the KFLANN, network parameter tolerance is defined based on the input feature itself. The first layer of the LVQ network is a competitive non-linear neural network that requires defining number of hidden units and learning rate. In SOM network three parameters need to be defined prior the learning process, there are error threshold, learning rate, controller for neighborhoods spread. LVQ and SOM network request to define their parameters prior the learning process base on the domain expert knowledge while KFLANN parameter setting is input data driven.

The comparison on the accuracy was recorded in Table 3-3. Matlab toolbox is used to test the LVQ, ART and SOM classification model.

Dataset	KFLANN	LVQ	ART	SOM
Iris	100.0	93.82	91.40	87.10
Thyroid	92.00	73.39	87.84	100.0
Glass	69.05	53.75	51.99	52.55
WBC	91.77	94.36	92.28	96.10
Wine	90.77	96.80	94.79	94.57
Heart	77.14	82.57	67.55	78.95
Segment	88.43	95.26	93.94	74.51
Waveform	77.40	89.57	74.16	81.40
Ionosphere	71.15	73.12	49.88	81.93
Dermatology	94.44	96.10	88.12	96.80
Spambase	86.01	81.46	84.49	85.79
Sonar	67.54	73.36	72.83	75.75

Table 3-3 Comparison on accuracy between KFLANN, LVQ and SOM

From the results, it was shown that KFLANN clustering algorithms performed comparable with the SOM, LVQ and ART clustering algorithms. Significance test according to non-parametric Mann Whitney U Test was performed between the pair of

clustering algorithm KFLANN versus LVQ, KFLANN versus ART and KFLANN versus SOM. This significance test does not assume any distribution of the tested dataset. The test was applied to independent samples which mean that each data pattern is independent of each other.

Compare the KFLANN and LVQ, the p-value obtained is 0.7075, this indicates that at significant value of 0.01 the difference between the prediction accuracy is statistically not significant.

KFLANN versus ART clustering algorithm, the p-value obtained is 0.6236. This value is greater than the cutoff value of 0.01. Thus it can be concluded that there is no difference between KFLANN and ART in prediction accuracy from statistical perspective.

p-value is 0.8174 when performed Mann Whitney test for the statistical evidence of a difference in prediction accuracy obtained using KFLANN and SOM. With the confidence level set at 95%, it can be said that there is insufficient evidence to support the differences is significant.

Experimental comparison between neural network clustering algorithms (KFLANN versus LVQ, ART and SOM) provided confidence that KFLANN performs as well as the other well established clustering algorithm.

A theoretical comparison between the KFLANN and other clustering algorithms is also briefly discussed in this thesis. Comparison with some famous clustering algorithms such as Resource Allocating Network (RAN) [99], Evolving Self Organized Map Network (ESOM) [23], [24] and Dynamic Evolving Neural-fuzzy Inference System (DENFIS) [59].

RAN model is a two layer network introduced by Platt. The first layer contains nodes that respond to a localized region of input features while the second layer aggregates outputs from first layer and creates the approximation function (of input-output mapping). It learns by allocating new units and adjusting the parameters of existing units whenever an unusual pattern is presented to the network. This feature is similar to the KFLANN output node creation without parameter adjustment. The network parameters are updated using standard LMS gradient descent paradigm, which is different from the KFLANN that only

involve centroids computation and re-shuffle of data points closest to centroids in its clustering process.

ESOM is an evolved variant of the Kohonen SOM [68]. There are three network parameters involving namely the error threshold, learning rate and control of neighborhood spread. Those parameters are user predefined based on domain expert knowledge. As there is a need for predefined parameters, this may be a costly trial-and-error process since the domain expert is not always available. This is not an issue in the KFLANN network since it uses data driven scheme of defining the network parameters. The ESOM network structure starts with an empty network and the nodes are created dynamically during the training process. This is similar to the KFLANN output node creation. A clear distinct topology map is created by ESOM which reduces feature dimensions and display similarities [68] while the KFLANN produces a series of output nodes that consolidate the groups of input data.

DENFIS is a fuzzy neural network that forms an area of partitioned regions (feature subspace) in the learning process. It consists of a more complex 5 layer architecture. The input layer integrates the input features into the system. The second layer is the layer where fuzzy quantization of input features is performed and third layer provides the rule units while the fourth layer performs fuzzy quantization of the output variables and fifth layer represents real values of the output variables. DENFIS employs hybrid learning scheme, it combines supervised and unsupervised learning scheme.

3.3 TOLERANCE INITIALIZATION USING THE STANDARD DEVIATION

In this section, the tolerance setting using the standard deviation is discussed in detail. The discussion covers two possible distribution that are normal distribution and non-normal distribution. The goal of this section is really to show why the standard deviation seems to provide a suitable heuristic means of obtaining the δ values.

3.3.1 Normal Distribution Cases

In the last section, the standard deviation method was proposed as a means to initialize the network. This section details the reasons for this suitability as a heuristic.

Figure 3.10 Normal distribution function curve

As is understood in the field of statistical, although there are many models of distribution, most data distributions in the natural world tend to fall into a typical bell-shaped curve which is also known as typical Gaussian distribution or Normal distribution. When the data is normally distributed, it means that most of the instances are close to the average while relatively few examples tend to one extreme or the other. Figure 3.10 shows the normal distribution curve and the coverage of the entire data range under different standard deviations is shown in the figure. According to the Empirical Rule [101]:

- About 68% of the data will fall in the interval between $\mu - \sigma$ to $\mu + \sigma$.
- About 95% of the data will fall in the interval between $\mu - 2\sigma$ to $\mu + 2\sigma$.
- About 99.5% of the data will fall in the interval between $\mu - 3\sigma$ to $\mu + 3\sigma$.

This rule is based on the normal distribution. The probability of any data point within the set X that is normally distributed, $P(\mu - \sigma < X < \mu + \sigma)$, form the z-scores

$$z_1 = \frac{\mu - \sigma - \mu}{\sigma} = -1, z_2 = \frac{\mu + \sigma - \mu}{\sigma} = 1 \quad (3-3)$$

The area under the normal distribution curve can be retrieved according the z_1 and z_2 , the area under the curve to the left of $z_1 = -1$ is 0.1586 while the area to the left of $z_2 = 1$ is 0.8413.

$$P(\mu - \sigma < X < \mu + \sigma) = 0.8413 - 0.1586 = 0.6827 \quad (3-4)$$

Hence the probability of any data point within the interval of $\mu - \sigma$ to $\mu + \sigma$ is 0.6827 (68%), the Empirical rule is sourced from this computations. Standard deviation measurement eventually indicates how tightly all the various instances are clustered around the mean of the dataset.

3.3.2 Non-Normal Distribution Cases

When the data is not the distribution of bell-shaped (normal distribution), the Empirical Rule mentioned is not applicable. The reference would now require the Chebyshev's Rule [98]. The Chebyshev's rule indicates that

- About 75% of the data will fall in the interval between $\mu - \sigma$ to $\mu + \sigma$.
- About 89% of the data will fall in the interval between $\mu - 2\sigma$ to $\mu + 2\sigma$.
- About 97% of the data will fall in the interval between $\mu - 3\sigma$ to $\mu + 3\sigma$.

Generally, Chebyshev's Rule states that there will be at least $1 - (1/k^2)$ within k deviation of the mean. Thus by the 6th standard deviation position, there would be at least mean yield of 97% of the data. In other words, if the δ value was set equal to σ , then it would require approximately 8σ to cover 100% of the data. This is rather similar to the 99.7% of the one stated in the Gaussian context. The subdivision of the space will therefore ensure that the entire feature space is broken into at least 6 portions. While the necessity to develop a higher resolution is not fully understood at this point of discussion, this subdivision is sufficient to ensure that the feature space is able to partition clusters such that the cluster formation is able to separate features.

For the normal distribution and non-normal distribution domain, it is suggested that the at least 6 standard deviation of the mean will approximately cover the entire data space. The

explanation is that for the normal distribution case, 6 standard deviations of the mean is able to cover the 99.7% of the entire data space and for non-normal distribution, 6 standard deviations cover 97% of the entire data space. With this, the standard deviation of the feature space can be used to initialize the feature tolerance and facilitate a heuristic means for the KFLANN to form relevant clusters and correctly represent class distributions. This heuristic approach is fundamental for the building of hierarchical structure in the chapters that follow.

In the KFLANN's tolerance setting, the standard deviation is the radii of the cluster and twice the standard deviation is the diameter of the cluster. Assuming that 6 standard deviation covers approximate the entire data space, it will form 3 to 4 well separable clusters. However this condition may not be sufficient to have proper cluster discrimination for certain problem sets. In cases where a more refined discrimination is required, a division of the standard deviation by a factor of 2 is used. This means that the cluster diameter is now 1 standard deviation and number of clusters will double to 6 or 8, which will likely provide a better discrimination power.

Thus, if the KFLANN is used as unsupervised clustering algorithm where the actual classes are unknown, the standard deviation can be used to initialize the tolerance value setting. This is ideal because the standard deviation of the feature spread can always be calculated. This computation can be performed for all features defined in the entire dataset. A further discussion of this is presented in Appendix A.

3.4 MAHALANOBIS DISTANCE BASED KFLANN

Another modification to the original KFLANN was made to improve the clustering capabilities. The distance measurement implemented in original KFLANN was the

generic Euclidean distance that functions by representing the square error between any 2 data points. Euclidean distance (ED) between two points is defined in equation (3-5).

$$ED(X_i, X_j) = \sqrt{(X_i - X_j)^T (X_i - X_j)} \quad (3-5)$$

$$ED(X_i, 0) = \sqrt{X_i^T X_i} = \|X_i\|_2 \quad (3-6)$$

In (3-6) data points with same distance from origin satisfy $\|X_i\|_2 = c$, if there are d -dimensional data points, the equation can be written as $X_1^2 + \dots + X_d^2 = c^2$ which is the equation for spheroid. It weighs all features equally with no specified biases. While this method is easy to implement and understand, it does not account for relationships between related variables nor can it focus and emphasis on dataset distribution related characteristics.

Another disparity measurement that is able to address the weaknesses of the Euclidean distance measure is the Mahalanobis distance. This distance metric can be used to replace the Euclidean distance metric in the KFLANN. P. C. Mahalanobis first introduced the Mahalanobis distance in 1936 [80]. It is a distance measurement that was based on the correlation between variables and takes into account the data set distribution. The Mahalanobis distance is useful as a means to determine the similarity of a set of variables (features) from any sample set. The Mahalanobis distance (MD) between data point X_i and X_j is computed using the equation:

$$MD(X_i, X_j) = |X_i - X_j|^T \text{Cov}(X) |X_i - X_j| \quad (3-7)$$

The covariance of data X , it is computed using the equation:

$$\text{Cov}(X) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^d \sum_{k=1}^d (x_{ij} - \mu_j) (x_{ik} - \mu_k) \quad (3-8)$$

A list of properties significant to the Mahalanobis distance measurement is [115]:

- Apart from the variances and covariance of the variables, it considers the mean values.
- It considers ranges of acceptability between features (variability).

- It accounts for interactions between features.
- It functions independently of the data dimensionality and is itself dimensionless.

3.4.1 Comparison between the Mahalanobis and Euclidean Distance

Measurement

The Mahalanobis distance seems to be a slightly better improvement over the Euclidean distance approach. This comparison can be viewed from different perspectives including the clusters distribution, covariance, variance, mean, scaling on the data sets and class decision boundaries.

Cluster Shape Comparisons

While the Euclidean distance method tends to describe a circular boundary around the cluster centroid (mean), the Mahalanobis distance considers the variance that yields an elliptic shaped cluster. In Figure 3.11, dash line ellipse circle is the cluster created using the Mahalanobis distance method. It weighs the variation in data set along the axis of projection. The Euclidean distance method on the other hand generates the spherical cluster center at z as shown in the figure.

Figure 3.11 Cluster created using Mahalanobis distance method (dash line) and the Euclidean distance method (solid line)

Data Treatment Comparisons

Another property that is advantages for the Mahalanobis distance method is the consideration of data set variability while Euclidean distance method does not. Instead of

treating all feature values equally in calculating the distance from the centroid, the Mahalanobis distance method imposes the weights through the differences by the range of variability in the direction of the data point. To a large extent, this is similar to Hebbian learning where there is a bias towards the major projection axis. Since Euclidean distance method does not take into account the variability of the feature values, it is therefore unable to provide an optimum discriminatory analysis.

Figure 3.12 Data distribution with data point A and B located from centroid of ellipse cluster, Z

In statistical inference, there is a preference to compute feature distances along with the variability of the variable. Feature values with high variability (variance) should weigh less than those with low variability. This fact is depicted in the Figure 3.12 and it can be noticed that if the Euclidean distance (ED) is used, $ED(A, Z) < ED(B, Z)$ (since it uses the sum of square difference between two points). However the Mahalanobis distance (MD) method yields a different conclusion of $MD(A, Z) > MD(B, Z)$ (since data point B located at the axis with the largest variability thus it weighs less and provides shorter distance from Z compared to data point A). The Euclidean distance method data point A and B will diagnose as a similar category since both have almost similar distance from the centroid Z. However data point A clearly lies along the elongated axis of the cluster, which in fact is not meant to be classified as the same group as data point B.

Data Point Scaling Comparisons

The Mahalanobis distance method automatically accounts for the scaling on the data points while the Euclidean distance does not impose this scaling property. Thus the

Mahalanobis distance is invariant to data point scaling which is an attractive advantage in the distance measurement. The standard deviation scaling makes Mahalanobis distance based clustering scale invariant.

Consider for example a two dimensional variable $Z = \langle Z_1, Z_2 \rangle$ with $Z_1 \in [0.01, 1]$ and $Z_2 \in [0.011, 0]$, it is obvious a large distance in the Z_2 dimension will create a larger Euclidean distance. However in the Mahalanobis distance, it expresses the distance between two values of a variable Z in the units of the standard deviation σ_z . This automatically accounts for the scaling issue.

$$MD(Z_1, Z_2) = \frac{|Z_1 - Z_2|}{\sigma_z} \quad (3-9)$$

Decision Boundary Comparisons

The Mahalanobis distance method is able to provide any arbitrary hyperplane as well as curve plane as the discriminant boundary base on the underlying data distribution. The Euclidean distance method is difficult (incapable) to represent on the curved decision boundary and represents the decision boundary as a linear hyperplane.

Figure 3.13 (a) Decision boundary created using Euclidean distance method and (b) Mahalanobis distance method

Figure 3.13 depicts the decision boundary created under Euclidean distance method and Mahalanobis distance method. The data points made up of 3 categories are differentiated by their contours. The Euclidean distance creates uniform spherical clusters as shown in the figure while Mahalanobis distance method creates elliptical clusters. The decision

hyperplane in the dash line illustrates that the Mahalanobis distance method is able to form curvicaudate boundary easier than the Euclidean distance method.

From the discussion above, the Mahalanobis distance implementation provides a more thorough treatment as compared with the Euclidean distance implementation but there is a price for these merits. The covariance computation requires certain extend of computational commitments. The memory and time requirement increases quadratically with the number of features. This is insignificant in low dimensional problems but may cause severe issues in higher dimensions.

3.4.2 Results on Mahalanobis Based KFLANN Experiments

Apart from the theoretical discussion provided to support the use of the Mahalanobis distance over Euclidean distance, the experimental testing was performed to evaluate the performance of Mahalanobis distance based KFLANN. The results are compared with the Euclidean based KFLANN in Figure 3.14. The simplest network setting was applied with standard deviation as feature tolerance and vigilance value was set to 0.9.

Figure 3.14 Difference between accuracy using the Mahalanobis distance method and Euclidean distance method on UCI datasets

For the tested benchmark datasets, it was shown that all tested datasets had better and if not comparable clustering accuracy in Mahalanobis based KFLANN than Euclidean based KFLANN.

From the characteristics studied on the Mahalanobis distance, it is noticed that Mahalanobis distance method is more applicable to datasets with correlated features. To verify the performance, an artificial dataset was created with an equal distribution of 500 data patterns and it consists of 10 feature dimensionalities. If a dataset has 2 correlated features, it means that two features are correlated and the remaining 8 features are not related at all. The Mahalanobis distance method and Euclidean distance method were used to compare the performance of the KFLANN on those artificial datasets.

Dataset	Number of correlated features	Fcorrelated/ d	Mahalanobis (%)	Euclidean (%)
AD6	2	0.2	100.0	95.65
AD7	3	0.3	100.0	94.87
AD8	4	0.4	100.0	94.41
AD9	5	0.5	99.38	94.41
AD10	6	0.6	98.76	86.96
AD11	7	0.7	100.0	89.44
AD12	8	0.8	92.55	83.23

Table 3-4 Comparison on clustering accuracy for the Mahalanobis distance method and Euclidean distance method on artificial datasets

Fcorrelated is the number of correlated features. The results were recorded on the artificial datasets in Table 3-4 with different portions of the correlated features from the entire data space. The difference between the accuracy obtained by KFLANN using the Mahalanobis distance method and Euclidean distance method (accuracy obtained using the Mahalanobis distance method minus the one using Euclidean distance method) was plotted in Figure 3.15 (a). The difference in accuracy divided by standard deviation is plotted in Figure 3.15 (b). As the ratio of *Fcorrelated/d* increases, the difference between the accuracies also increases. A significant difference between the accuracies is registered when there is a ratio larger than 0.5. This is shown in Figure 3.15 (b) that artificial

datasets with $F_{correlated}/d$ ratio 0.5 and above the difference in accuracy in the standard deviation unit are significant at confidence interval at least 98% (the length of the bar is greater than 2).

Figure 3.15 (a) Difference between accuracy using the Mahalanobis distance method and Euclidean distance method on artificial datasets and (b) divided by standard deviation

From this illustration, it can be concluded that Mahalanobis distance method is useful when a degree of correlation is resident amongst features. The cost for covariance computation is proportional to the feature dimensionality. If the domain is known not to possess inter-feature relationships, then the Euclidean distance method is sufficient for the reasons of computational efficiency.

3.5 CHAPTER SUMMARY

In this chapter, the KFLANN's network parameters consisting of the vigilance and tolerance setting were presented. The vigilance setting was shown to be linearly correlated to the number of clusters. As such, when the vigilance value increases, the number of clusters will increase and vice versa. Although, the tolerance setting using the standard deviation method is able to obtain optimal performance for the algorithm, it cannot be substantiated as the only method or the best way to initialize the tolerance value. Tolerance tuning is an optional step when the clustering outcome does not achieve expectation. However, in the cases investigated, the standard deviation initialization

suffices for network convergence and good clustering results. Experimental analysis showed that smaller tolerance values tend to create a larger number of clusters.

The consistency of the KFLANN algorithm has been studied. Results show that the proposed data pattern reshuffling method is able to provide consistent performance that solves the inconsistency issues in clustering. From the experimental investigation, the modified KFLANN creates a stabilized set of clusters within small number of iterations. This implies that it is efficient in the perspective of computational complexity.

Comparative study between KFLANN and some well-known neural network based clustering algorithms was performed in section 3.2.2. Theoretical comparison between KFLANN with non neural network based clustering algorithms indicated that KFLANN indeed encapsulate some of advantageous characteristic of the compared algorithms. The experimental results showed that the KFLANN is capable to perform comparable with LVQ, ART and SOM.

The KFLANN clustering algorithm like majority of the clustering methodologies needs to adopt a distance measurement strategy. In this chapter, the Mahalanobis distance method is compared to the Euclidean distance method using in the KFLANN algorithm. Theoretical analysis and experimental results show that Mahalanobis distance method boosts the accuracy when the domain contains correlated features. If the domain does not have any correlated features, both Mahalanobis distance method and Euclidean distance method will achieve the relatively similar accuracies.

4 HIERARCHICAL FAST LEARNING ARTIFICIAL NEURAL NETWORK

Classifier systems tend to be hindered by the curse of dimensionality. The KFLANN clustering algorithm also faces the same fate, though to some lesser degree. In order to extent the scalability of KFLANN, a hierarchical network consisting of localized KFLANN networks was designed to combat the curse of dimensionality. The resultant hierarchical network is known as the Hierarchical Fast Learning Artificial Neural Network (HieFLANN).

4.1 INTRODUCTION

In this chapter, three strategies were significant to the development of the HieFLANN.

These include the following:

- A self-organized approach for network construction with a definitive initialization process.
- Clustering of the input features with a similarity measurement to create a series of sub-networks (subnet).
- Compression of clusters created from sub-networks into transformed (reduced) feature spaces.

The algorithm begins by extracting the statistical distribution of individual features and obtains the network initialization parameters for the KFLANN. After the parameters are determined, the KFLANN is used to segment the regions and separate features into similarity clusters. These clusters then become the groupings by which the global input space is to be segmented. The network initialization of the HieFLANN can be achieved automatically using the statistical distribution of the feature values. The clustering of feature similarity is a non-parametric approach that requires no assumption on the data set distribution. Feature similarity, through the Mahalanobis distance measurement,

segregates the various input feature spaces. The KFLANN clustering of the features produces the output Homogenous Feature Space (HFS) that determines the number of the KFLANN subnets to be built at the subsequent layer.

Within each HFS, the KFLANN again used to perform localized clustering. This clustering will result in collection of KFLANN subnet clustering outputs and the corresponding links to the input data. The pairing of these input and output data provides the means to derive a transformation matrix that can be further used for data compression. The rationale to have feature compression on the raw data set is to mitigate the severe effects of the “curse of dimensionality”.

The CCM represents the correlation coefficient of the input data and the clustering results of all subnets. With the use of eigen-decomposition, the eigenvectors are derived from the CCM. These eigenvectors are then used for data compression. After the data compression procedure, the reduced feature dimensionality is used as input data to the top-most KFLANN unit. This KFLANN no longer receives the original feature space inputs but has the compressed data received from clustering. Results indicate that the proposed strategy is successful in reducing the dimensional space and continue to maintain the original integrity in the clusters.

4.2 HIEFLANN ARCHITECTURE

This section presents the hierarchical structure of the KFLANN and the rationale for the various components. It highlights the use of an unsupervised paradigm to build the self-organizing network. The initial network construction requires the self-organizing stage that attempts to discover “natural categories” embedded in the dataset. These discovered ‘groupings’ are subsequently used to aid the segmentation of data features into categories,

and through hierarchical techniques can be combined to form the solution for the problem.

A sample structure of HieFLANN is shown in Figure 4.1. The KFLANN subnets are determined from the initial HFS construction stage. More details of the structure are provided in the next few sections as the presentation dwells deeper into the structure and concept of the entire framework.

Figure 4.1 HieFLANN architecture

4.3 HIEFLANN ALGORITHM

The pseudocode of HieFLANN is shown below. Details of each step are subsequently presented.

<i>Input: Data, Output: Data Clustering</i>	
<i>BEGIN</i>	
1 Create homogenous feature subspace	<i>Pre-processing</i>
2 Perform KFLANN clustering on every homogenous feature subspace created	<i>Local Distribution Analysis</i>
3 Form canonical correlation matrix	<i>Feature Transformation</i>
4 Perform eigen-decomposition on the canonical correlation matrix	
5 Compress feature space	
6 Perform KFLANN clustering using the compressed feature space in Step 5	<i>Classification</i>
<i>END</i>	

Figure 4.2 HieFLANN pseudocode

4.3.1 Homogenous Feature Subspace Generation, a First Stage

Clustering

The initial pre-processing of the HieFLANN involves the preparation of homogenous feature subspaces (HFS) for subnet processing. The HFS is a basic representation of a cluster of features bearing similar feature types. Note that this is a cluster formation of the input feature types and not the actual data that populated the features. Input features within the same HFS are more similar, as compared to features of other HFS. The outline of HFS generation follows.

```

Input: Data
Output: HFSs
BEGIN
1  Compute features similarity
2  Form the feature similarity matrix
3  KFLANN clustering on the feature similarity matrix
END

```

Figure 4.3 Pseudocode for HFS generation

The HFSs are created based on the similarity profile across the features in the entire input feature space. The Mahalanobis distance (MD) in equation (4-1) is used to compute the feature similarity amongst all features of the input feature space.

$$MD(X_i, X_j) = (X_i - X_j)^T * Cov(X) * (X_i - X_j) \quad (4-1)$$

X_i is the i^{th} feature, $*$ is the matrix multiplication symbol.

The covariance takes the feature space variance into consideration before the Mahalanobis distances are calculated. Different weights can be assigned to different features base on the variance, making this an attractive advantage for distance measurement. Due to the symmetrical characteristic of feature similarity, the complexity of preparing the feature similarity profile is $d(d-1)/2$. The feature similarity matrix consists of numerical values measuring feature similarities and these are used as inputs to the KFLANN clustering algorithm. The output classes of the feature clustering become the corresponding HFSs. This is illustrated in Figure 4.4.

Figure 4.4 Construction of homogenous feature subspace

The creation of the HFSs is essentially to reduce the effect of redundant features. These HFS groupings are used in subsequent processes in HieFLANN as an attempt to ease subnet analysis since features within HFS are closer to each other compared to features residing in other HFSs.

The HFS construction is an essential step to ensure smooth processing at the subsequent hierarchical layers. Without the HFS step, it may not be possible to perform successful fusion of clusters at the later stages of the hierarchical network. Such a situation can occur if the subnets are poorly selected. The HFS construction process is statistically driven and is an autonomous procedure that requires no externally defined parameters. The KFLANN needs to determine the vigilance and tolerance value and the process of network initialization is given in an earlier discussion in Chapter 3. The feature tolerance value setting is set according to the statistical standard deviation. In order to obtain an optimal outcome with the best discriminative power, the KFLANN clusters feature similarity profile must be performed in a tight discrimination mode such that the vigilance is near unity. This is to ensure that tighter HFSs are created.

The initial HFS construction determines the number of KFLANN subnets needed in the hierarchical structure. This implies that the HieFLANN can be designed as a data dependent network which forms the network structure autonomously based on statistical

information of the problem domain. The HSFs are groups of similar features that will be partitioned further using KFLANN clustering algorithm into sub-groups.

4.3.2 Local Distribution Analysis, a Second Stage Clustering

In the previous section, the method to identify the HFS was presented. This section continues with the subsequent process that implements subnets of mutually exclusive KFLANNs using the discovered HFSs. For example, if the first HFS contains of feature 1, feature 3 and feature 4, then the first KFLANN subnet will be given the subset of the input contain feature 1, feature 3 and feature 4. Each subnet capsules input features with similar characteristics and performs a localized clustering of the data. Note that each of these KFLANN processes is mutually exclusive between subnets and thus a parallelization of this portion is possible. This parallelization will enable the model to speed up processing and increase processing efficiency. The local distribution analysis is performed using KFLANN clustering analysis on the presented feature subspace.

4.3.3 Feature Compression

A common problem that incapacitates many network models is the inability to perform learning in high dimensionality of feature spaces. To achieve the desired classification performance, the feature dimensionalities are often reduced, compromising the accuracy of representations in the original space. Thus, efforts are made to carefully lower the feature dimensional space. The use of the locally lower dimensional feature space can often be adequate if the compression of the feature space is performed correctly. This is essentially the key to this proposed HieFLANN. This thesis proposes the use of a method that is referenced as Canonical Covariance Feature Compression.

The proposed Canonical Covariance Feature Compression (C2FeCom) requires the eigen-decomposition on the Canonical Correlation Matrix (CCM). To begin this process, the data matrix containing the collection of all the subnet input and corresponding output is prepared. The correlation coefficient computation is then performed on the data matrix to generate CCM.

An example is given to instantiate the feature compression process presented. For simplicity of illustration, assume that 2 subnets were created on a set of data with $d=5$. Further assume that the HFS1 (subnet 1) and HFS2 were clustered with 3 and 2 features respectively. The CCM generated is shown in Figure 4.5.

Subnet 1				Subnet 2		
X ₂	X ₄	X ₅	Y ₁	X ₁	X ₃	Y ₂
1	0.4	1.1	2	0.6	0.7	2
0.5	0.8	0.4	1	0.5	0.2	1
0.4	0.5	0.3	1	0.1	0.2	1
0.7	1.0	0.8	2	0.8	0	3
0.8	0.2	0.2	2	0.7	0.3	1

Combined matrix						
X ₂	X ₄	X ₅	X ₁	X ₃	Y ₁	Y ₂
1	0.4	1.1	0.6	0.7	2	2
0.5	0.8	0.4	0.5	0.2	1	1
0.4	0.5	0.3	0.1	0.2	1	1
0.7	1.0	0.8	0.8	0	2	3
0.8	0.2	0.2	0.7	0.3	2	1

CCM	X ₂	X ₄	X ₅	X ₁	X ₃	Y ₁	Y ₂
X ₂							
X ₄							
X ₅							
X ₁							
X ₃							
Y ₁							
Y ₂							

CCM =	R _{xx}	R _{xy}
	R _{yx}	R _{yy}

Figure 4.5 Example for CCM construction

The elements of CCM were computed using the Pearson's correlation coefficient. These equations are represented from (4-2) to(4-4).

$$R_{X_i X_j} = \frac{Cov(X_i, X_j)}{\sigma_{X_i} \sigma_{X_j}} \quad (4-2)$$

$$R_{Y_p Y_q} = \frac{Cov(Y_p, Y_q)}{\sigma_{Y_p} \sigma_{Y_q}} \quad (4-3)$$

$$R_{X_i Y_p} = \frac{Cov(X_i, Y_p)}{\sigma_{X_i} \sigma_{Y_p}}, \quad R_{Y_p X_i} = R_{X_i Y_p}^T \quad (4-4)$$

Given

$$Cov(X_i, X_j) = \frac{1}{n} \sum (X_i - \mu_{X_i})(X_j - \mu_{X_j})^T, \quad \sigma_{X_i} = \sqrt{\left(\frac{1}{n} \sum (X_i - \mu_{X_i})^2 \right)}$$

$1 \leq i \leq d, \quad 1 \leq j \leq d, \quad 1 \leq p \leq S, \quad 1 \leq q \leq S$, Y_p is the p^{th} subnet clustering encoding.

Cluster encoding usually takes the cluster index directly.

R_{xx} , R_{yy} , R_{xy} and R_{yx} are in matrix form. The correlation between features variables is captured in R_{xx} . The correlation between features and cluster grouping attributes is represented in R_{xy} and its transpose R_{yx} . The correlation between cluster grouping attributes is valued in R_{yy} that representing the association amongst the cluster grouping attribute created in each subnet.

The eigen-decomposition is performed on the CCM square matrix of $S+d$ by $S+d$, which then retrieves the eigenvector and eigenvalue pair. The eigenvector of a matrix is a vector that when multiplied with the matrix will result an integer multiple of that original vector (similar to ‘factor’ of matrix). This integer is the corresponding eigenvalue of the eigenvector.

The description of the exact algorithm for determination of eigenvectors and eigenvalues is omitted here as it can be referenced from standard mathematical programming tools. In this thesis, the eigenvectors and eigenvalues are computed using an available mathematical software library. Eigenvectors obtained from the above computation are used for the proposed C2FeCom feature compression process. It involves canonical covariance analysis. The detailed illustration of C2FeCom is given later. Let u be the

eigenvector that is obtained from the eigen analysis on the CCM. The compression is formularized by equation (4-5) and graphically illustrated in Figure 4.6.

$$C2FeCom(X) = X * u \quad (4-5)$$

Figure 4.6 Pictorial view on C2FeCom process

4.3.4 Final Convergence of Information, a Third Stage Clustering

The last procedure within the HieFLANN is another clustering process that makes use of the KFLANN to cluster the compressed feature space and prepare the labeling of each class to instances in the problem domain. The performance of HieFLANN is then measured by comparing the group assignment to the actual class representation. This is to verify the accuracy between the reduced feature space representing the class distribution and the original feature space. Results of these are shown later.

4.4 CANONICAL COVARIANCE FEATURE COMPRESSION IN HIEFLANN

The stages of the HieFLANN provide a systematic procedure, funneling the data initially into the different HFSs. After the second stage of clustering, individual HFSs will form clusters that are characterized by the specific features found within the HFS. This work focuses on how the feature compression for the problem space can be accomplished on

the basis of canonical covariance analysis. The proposed Canonical Covariance Feature Compression process is depicted in Figure 4.7.

Figure 4.7 Preparation process for C2FeCom

The example in Figure 4.5 is used to illustrate the C2FeCom process. In this example, let $X = \{X_1, X_2, X_3, X_4, X_5\}$, be the input feature space of dimensionality 5. Assume that the input feature space is divided into 2 HFSs namely HFS1 and HFS2. HFS1 may consist of $\{X_2, X_4, X_5\}$ and HFS2 consists of $\{X_1, X_3\}$. In this example there are two KFLANN subnets, the first subnet deals with HFS1 resulting Y_1 , (represent the cluster index) while second subnet handles HFS2 output Y_2 . Let u and v be the basic vector to be determined such that the correlation between $\hat{X} = v^T X$ and $\hat{Y} = u^T Y$ is maximized. $\hat{X}$ is the transformation of X under the vector v , $\hat{Y}$ is the transformation of Y by vector u . The problem task can be formulated as optimization problem as follow:

$$\begin{aligned} & \text{Maximize } u^T R_{xy} v \\ & \text{subject to } v^T R_{yy} v = u^T R_{xx} u = 1 \end{aligned} \quad (4-6)$$

Let α and β be the Lagrange multipliers [7], formulated the above optimization function in Lagrange equation as follow:

$$f = u^T R_{xy} v - \frac{1}{2} \alpha (u^T R_{xx} u - 1) - \frac{1}{2} \beta (v^T R_{yy} v - 1) \quad (4-7)$$

The subsequence procedures are Lagrange equation solutions. Apply differentiating f with respect to u and v ,

$$\frac{\partial f}{\partial u} = R_{xy}v - \alpha R_{xx}u = 0 \quad (4-8)$$

$$\frac{\partial f}{\partial v} = R_{yx}u - \beta R_{yy}v = 0 \quad (4-9)$$

Multiply (4-8) with u^T

$$u^T R_{xy}v - \alpha u^T R_{xx}u = 0 \quad (4-10)$$

Multiply (4-9) with v^T

$$v^T R_{yx}u - \beta v^T R_{yy}v = 0 \quad (4-11)$$

Subtract (4-10) from (4-11)

$$v^T R_{yx}u - \beta v^T R_{yy}v - u^T R_{xy}v + \alpha u^T R_{xx}u = 0 \quad (4-12)$$

Together with the constraints and (4-12), $\alpha - \beta = 0$, thus $\alpha = \beta$. Substitute this into (4-9) to obtain

$$R_{yx}u - \alpha R_{yy}v = 0 \quad (4-13)$$

From (4-13),

$$v = \frac{1}{\alpha} R_{yy}^{-1} R_{yx}u \quad (4-14)$$

Substitute (4-14) into (4-8) gives

$$\begin{aligned} R_{xy} \left(\frac{1}{\alpha} R_{yy}^{-1} R_{yx}u \right) - \alpha R_{xx}u &= 0 \\ R_{xx}^{-1} R_{xy} R_{yy}^{-1} R_{yx}u &= \alpha^2 u \end{aligned} \quad (4-15)$$

Let $A = R_{xx}^{-1} R_{xy} R_{yy}^{-1} R_{yx}$, then

$$Au = \alpha^2 u$$

Equation (4-15) is in the form of the eigenproblem $Ax = \lambda x$. The solution for the described optimization problem is the solution to eigenproblem that involves the determination of the eigenvector and eigenvalue. The u is then used for the compression proposed in (4-5). It defines a transformation of X in the direction of u .

4.4.1 Rationale of Proposing C2FeCom

The Canonical Correlation Analysis (CCA) determines the statistical significance of multivariate relationship between sets of variables. It is the only known multivariate approach that handles multiple dependent variables. In addition, CCA is also the only multivariate method that works with both dependent and independent variables

simultaneously. Other motivations for adopting the CCA include the fact that canonical variate can be interpreted as a level of significance (statistical and practical significance). Additional work may involve interpreting the canonical variates produced during the decomposition. This however, is beyond the scope of this thesis.

The Correlation Coefficient, is related to covariance in the formulation:

$$CorrCoef(X, Y) = \frac{Cov(X, Y)}{\left(\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)^2} \right) \left(\frac{1}{n} \sqrt{\sum_{i=1}^n (y_i - \mu_y)^2} \right)} \quad (4-16)$$

Thus CCA too enjoys the merits provided from covariance term. Some of the benefits include:

- CCA is invariant to affine transformation and this is a desirable property for pattern classification [12]. Correlation coefficient elements of CCM are symmetrical in nature that ensures consistency regardless the order of feature values.
- Correlation accounts for ranges of acceptability between features (variability). This mitigates the effect of feature values that falling out of the acceptable range.
- The covariance imposes weights that are distributed in the feature space. This eliminates the co-linearity effect of the dataset.

4.5 PERFORMANCE ANALYSIS OF THE HIEFLANN

To verify the performance of the proposed HieFLANN with its feature compression method C2FeCom, benchmark datasets from the UCI Repository of the machine learning database [85] were used. There were 3 categories of datasets used in this performance analysis, there were the low-dimension dataset with number of dimension d defined as, $1 < d \leq 10$, medium-dimension dataset, $10 < d \leq 50$ and high-dimension dataset, $d > 50$. The dataset was divided into 2 portions namely the training set and test set. The training data set was used for train the network while test set is used to verify the generalization performance.

The *ROC* was adopted to organize the performance measurement of the HieFLANN classification model. The *R* was also used in the result analysis. Detail information of *ROC* and *R* are given in Appendix B.

4.5.1 Experimental Result And Discussion On The UCI Datasets

Results for classification of low dimensionality dataset are listed in Table 4-1. The HieFLANN was compared with the classical KFLANN representing the non hierarchical (single network architecture) clustering model.

Data set	d	S	HieFLANN (%)	KFLANN (%)	R
Iris	4	1	95.24	100.0	0.9545
Thyroid	5	3	95.00	92.00	1.0285
Glass	9	4	73.67	69.05	1.0365
WBC	9	4	97.23	91.77	1.0531

Table 4-1 Classification accuracy for low dimensionality dataset

The results showed that for low-dimensional datasets, HieFLANN has better predictive accuracy. Most of the datasets in this category consistently showed a higher accuracy in HieFLANN than in the KFLANN. The error ratios in the last column is greater than 1 indicate that the reduced feature space preserved the original class discrimination while retaining the prediction performance. These results indicated that the compression implicit in the HieFLANN can help to improve the accuracy on the low dimensionality datasets.

Data set	d	S	HieFLANN (%)	KFLANN (%)	R
Wine	13	6	96.67	90.77	1.0571
Heart	13	5	87.53	77.14	1.0923
Segment	19	4	82.26	88.43	0.9484
Waveform	21	13	81.96	77.40	1.0386
Ionosphere	33	7	83.40	71.15	1.1050
Dermatology	34	19	85.40	94.44	0.9211

Table 4-2 Classification accuracy for medium dimensionality dataset

For medium-dimensional datasets, similar outcomes that indicate better performance in the HieFLANN were obtained. The HieFLANN has a better accuracy for most datasets in

this category except the Segment and Dermatology dataset. The Segment and Dermatology data had an R value 0.9484 and 0.9211 respectively. Since the value was close to 1, the compressed feature spaces for these two domains still reflect the distribution of original feature space.

Data set	d	S	HieFLANN (%)	KFLANN (%)	R
Spambase	57	28	83.14	86.01	0.9754
Sonar	60	18	75.19	67.54	1.0613

Table 4-3 Classification accuracy for high dimensionality dataset

For high-dimensional datasets, the clustering results of the HieFLANN were found to be comparable to the KFLANN as shown in Table 4-3.

Figure 4.8 Accuracy comparison between HieFLANN and KFLANN on UCI datasets

Figure 4.8 is the plot that comparing the performance of the HieFLANN and the KFLANN for the three groups of datasets. The HieFLANN is able to mitigate the irrelevant, redundant feature encapsulated in the original feature space.

Figure 4.9 Difference in accuracy between HieFLANN and KFLANN in the standard deviation unit

Figure 4.9 shows the difference in accuracy of the HieFLANN and KFLANN (accuracy of HieFLANN minus KFLANN) in standard deviation unit. The results of the HieFLANN

and two other techniques are tabulated in Table 4-4. This represents an accuracy comparison with two well accepted methods. The HieFLANN performed better for 8 out of 12 datasets. Subsequent graphical plots were used to illustrate the tabular comparisons in Table 4-4. A plot of the difference in accuracy divided by the standard deviation was done for each type of dataset. Thus, when the length of the bar on the standard deviation chart is higher than 2, this indicate that the results are significant at the 95% confidence interval.

Further investigations on the HieFLANN were carried through the comparison with other classification methodologies such as the Radial Basic Function (RBF) and C4.5. These methodologies were supervised clustering algorithm. Weka [52] implementation of C4.5 and RBF was used for this comparison.

No	Data set	HieFLANN (%)	RBF (%)	C4.5 (%)
1	Iris	95.24	91.90	95.30
2	Thyroid	95.00	95.48	85.33
3	Glass	73.67	62.86	65.71
4	WBC	97.23	96.10	74.50
5	Wine	96.67	87.70	93.80
6	Heart	87.53	71.67	78.33
7	Segment	82.26	62.61	91.08
8	Waveform	81.96	73.45	77.31
9	Ionosphere	83.40	84.15	77.63
10	Dermatology	85.40	75.39	96.37
11	Spambase	83.14	61.20	85.37
12	Sonar	75.19	57.90	70.43

Table 4-4 Comparison on the accuracy obtained from HieFLANN, RBF and C4.5

(a) HieFLANN versus RBF

(b) HieFLANN versus C4.5

Figure 4.10 Comparison of the accuracy difference between HieFLANN, RBF and C4.5

Figure 4.10 plotted the different between accuracy obtained by the pair of compared algorithm, (a) is the plot of different in accuracy and accuracy in the standard deviation unit for HieFLANN compared to RBF while (b) is the comparison between HieFLANN and C4.5.

In the HieFLANN versus the RBF comparisons, there were 10 datasets that the HieFLANN performed better than RBF algorithm. It was noticed that 3 out of the 10 datasets had the results significant at the 95% confidence interval. Another 5 out of the 10 datasets had the significance graded at least 68% confidence interval.

The comparison between the HieFLANN and the C4.5 clustering yielded 4 out of 10 of datasets showing that the HieFLANN cluster performed better with at least 68% confidence interval.

4.5.2 C2FeCom Revisited - HieFLANN Clustering Using the Wine

Dataset

The Wine dataset was used as means to explain how the HieFLANN performs clustering.

4.5.2.1 First Stage of Clustering, HFS Generation

The characteristics of Wine were represented such that it contains 13 features known as F1 to F13. In the first stage of the HFS construction, 6 HFSs were created by KFLANN clustering algorithm. This formation was based on the feature similarity extracted and shown in Table 4-5.

HFS	Feature
1	F1
2	F2, F10
3	F3, F4
4	F5, F13
5	F6, F7, F9, F11, F12
6	F8

Table 4-5 HFSs for Wine dataset

The network structure for this dataset consisted of 6 KFLANN subnets as shown in Figure 4.11.

Figure 4.11 HieFLANN structure for Wine dataset

These 6 HFSs grouped similar features together. Features that are grouped together within the same HFS imply a higher similarity with feature within the HFS compared to features found in other HFSs.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
F1	0	3.5265	2.4377	3.3274	3.0742	2.9927	3.4508	3.6389	3.2708	2.8808	3.6668	3.6584	2.8335
F2	3.5265	0	3.4945	3.1034	3.0355	3.9948	3.9478	3.1613	3.3664	2.7432	4.0411	4.5926	3.7699
F3	2.4377	3.4945	0	1.8543	3.0593	2.7382	3.2902	2.7238	3.0357	3.225	3.2682	3.1794	3.482
F4	3.3274	3.1034	1.8543	0	3.0487	3.2044	3.5138	2.31	2.9357	3.1974	3.2329	3.6117	3.8953
F5	3.0742	3.0355	3.0593	3.0487	0	2.8164	2.4818	3.5246	2.1509	2.4185	2.7098	3.6234	2.3136
F6	2.9927	3.9948	2.7382	3.2044	2.8164	0	1.5685	4.0266	2.0361	3.7455	2.4726	1.9956	2.824
F7	3.4508	3.9478	3.2902	3.5138	2.4818	1.5685	0	4.1701	1.8566	3.6672	2.0759	2.2492	2.5294
F8	3.6389	3.1613	2.7238	2.31	3.5246	4.0266	4.1701	0	3.481	3.2885	3.614	4.3631	4.1518
F9	3.2708	3.3664	3.0357	2.9357	2.1509	2.0361	1.8566	3.481	0	3.1672	2.2922	2.7953	2.7553
F10	2.8808	2.7432	3.225	3.1974	2.4185	3.7455	3.6672	3.2885	3.1672	0	3.9343	4.7632	2.8016
F11	3.6668	4.0411	3.2682	3.2329	2.7098	2.4726	2.0759	3.614	2.2922	3.9343	0	2.5122	2.9048
F12	3.6584	4.5926	3.1794	3.6117	3.6234	1.9956	2.2492	4.3631	2.7953	4.7632	2.5122	0	3.6601
F13	2.8335	3.7699	3.482	3.8953	2.3136	2.824	2.5294	4.1518	2.7553	2.8016	2.9048	3.6601	0

Figure 4.12 Feature similarity

The feature similarity is given in Figure 4.12 and this was fed to the KFLANN clustering algorithm. This feature similarity matrix of size 13×13 ($d \times d$) is similar to the one with the dataset of 13 features and 13 data patterns. Feature tolerance is computed using the standard deviation method and displayed in Table 4-6.

Feature	Tolerance
F1	0.9707
F2	1.1092
F3	0.9443
F4	1.0103
F5	0.9066
F6	1.0963
F7	1.1609
F8	1.1418
F9	0.9307
F10	1.1026
F11	1.0595
F12	1.2816
F13	1.0472

Table 4-6 Feature tolerance for Wine 's feature similarity data

4.5.2.2 Second Stage Clustering, Local Distribution Analysis

6 KFLANN subnets were then created to execute in parallel within individual HFS as shown in Figure 4.13. The network initialization for each KFLANN was achieved using the standard deviation method as the feature tolerance and the vigilance was set to a value of 0.9.

Figure 4.13 6 KFLANN subnets with the respective output form the R_{yy}

Each KFLANN subnet produced a mutually exclusive clustering outcome that was represented in y_i , where i is the subnet index. Let Y be the matrix that forms from each individual vector of y_i , for the Wine dataset, 6 HFSs yield matrix Y of size $n \times 6$. KFLANN subnet clustering output follows the simple index sequence encoding of the cluster formation.

4.5.2.3 Third Stage, Feature Compression

Correlation coefficient is then computed on the cluster along with the correlation coefficient elements computed on the entire feature space as shown in Figure 4.14.

Figure 4.14 Correlation coefficient on the feature space form the R_{xx}

The Canonical Covariance Matrix is then derived and it consists of the R_{xx} , R_{yy} , R_{xy} and R_{yx} . After this stage, the matrix is ready for the C2FeCom process.

Figure 4.15 Correlation coefficient on the feature space and KFLANN subnets clustering output form the R_{xy}

Using the proposed $C2FeCom = X*u$ where u is the eigenvector. The eigenvector is obtained from the eigenanalysis on the CCM.

4.5.2.4 Final Stage, Fusion of Information

The feature space transformation created a transformed feature space of the size $n \times 6$ for the last stage KFLANN clustering. The final stage of the KFLANN clustering follows the standard setting for the network parameter setting that are vigilance set as 0.9 and the standard deviation is used for the feature tolerance setting.

4.6 DISCUSSION ON THE HIEFLANN

The HieFLANN is a massively parallel-connected artificial neural model consisting of simple processing KFLANN subnet modules. These subnets interact extensively with eventual connections further along the process. In a way, this mimics the way biological nervous systems seem to operate. The subnet interactions are realized thru the feature compression phase. Localized groupings were first performed at each KFLANN subnet and the cluster representations were then feedforward for the compression process.

It is a self-organized artificial neural model such that the network construction is an automated process that is data driven. No stringent parameters setting were needed. The only parameters were the tolerance and vigilance in KFLANN and these can be determined according to the statistical information extracted from the data. Thus it is a flexible neural model for pattern classification.

The compression method implicit in the HieFLANN provides excellent discrimination. This was illustrated by the experimental evaluation on the various benchmarking problem sets. The HieFLANN seems to work well for domains with large feature dimensionalities as reflected in the results obtained.

The HieFLANN exhibits the stability and reliability of clustering from its KFLANN subnets. The cluster consistency in the KFLANN has been shown in Chapter 3 with its data pattern reshuffling property. The consistent clusters created insure the stabilization of the feature compression resulting promising results.

Competitive learning that encapsulated within subnets of the HieFLANN is beneficial to the classification process. The KFLANN subnets layering within the HieFLANN exhibit the Winner Take All (WTA) competitive learning process. The competitive layers are able to assess the occurrence of incomplete feature. The feature provides the HieFLANN with fault tolerance ability.

The idea of developing the hierarchical based ANN with modular implementation does not impose any restriction on the type of subnet resident within the hierarchical structure. The stability of KFLANN unsupervised clustering algorithm is the main reason of chosen it as the subnet structure within the HieFLANN.

Despite the merits provided by HieFLANN, it still faces several limitations. At this moment it only account for continuous value feature. The CCM must be in numerical form in order to retrieve the eigenvectors. In order to incorporate other types of feature such as nominal and ordinal, it is possible to look into the correlation measurement on the corresponding feature type.

C2FeCom of HieFLANN is only capable of extracting linear correlations between variables. It however cannot account for nonlinear domains. It is not easy to analysis nonlinear correlation coefficient. It is easier to extract nonlinear correlations using information theory. There is a strong relation between correlation and mutual information.

4.7 CHAPTER SUMMARY

In this chapter, the proposed hierarchical network structure of the KFLANN subnet was discussed. The rationale for developing the HieFLANN was described in detail. The feature compression method adopted in the HieFLANN was based on the Canonical Covariance Analysis that transformed the original feature space onto the smaller feature dimensionality space (reduced). The eigen analysis was used to derive the pair wise canonical variates for the compression purpose. Explanation on the CCA using an exemplar was used to illustrate the optimization problem can be transformed into eigen problem. The solution to the eigen-problem yielded the eigenvectors with the respective eigenvalues, the eigenvector was subsequently used for the compression formulation.

Experimental evaluation on the HieFLANN using real life datasets were performed, and result comparisons with other methodologies was also carried out. It is evident that the totally unsupervised HieFLANN network is able to cluster the given dataset with

relatively good accuracies. The HieFLANN network performs clustering and feature space compression in a hierarchical bottom up processing manner.

The HieFLANN can also function effectively on high dimensionality problem domains, however it suffers from some limitations. The eigen-analysis essentially involved the rotation computation that requires an enormous amount of memory and at the same time involves heavy computations. Although the computational complexity involved in the canonical covariance analysis is not detailed in this thesis, the computational involvements can be deduced to be intensive. The computational complexity for eigen analysis is largely ignored due to the efficacy of the existing algorithms to handle it.

5 HIERARCHICAL HYBRID LEARNING MODEL

K-ITERATIONS FAST LEARNING ARTIFICIAL

NEURAL NETWORK AND

BACKPROPAGATION

This chapter uses the concepts in Chapter 3 and 4 to build hierarchical architectures that can help overcome the curse of dimensionality. An extension of the model into the hierarchical approach is examined. This chapter builds on the Homogenous Feature Subspaces (HFSs) principles described in Chapter 4.

5.1 THE NEED OF HIERARCHICAL HYBRID LEARNING

MODEL

After obtaining positively significant results from the hierarchical strategy in the HieFLANN clustering ANN model, a second hierarchical model is examined. This model is computationally less intensive and uses the hybrid learning model described later. In Chapter 4, the HieFLANN model used individual KFLANNs to form localized clusters, which were subsequently compressed using C2FeCom. The final stage fuses the segmented feature space using yet another KFLANN network. In the hierarchical hybrid learning model developed in this chapter, the initial feature separations are again obtained using the HFSs. Instead of using the C2FeCom, another method involving a hybrid of the KFLANN with MLP-BP type networks is introduced. To extend the hybrid model, the resultant hybrid models were hierarchically cascaded, with each later layer bearing the

responsibility of fusing outputs from previous layers together. This is illustrated in Figure 5.1. This hierarchical process continues until the final output layer is reached.

5.2 ALGORITHM AND ARCHITECTURE OF HIEFLANN_BP

This chapter illustrates how the HFS formation first segments a large problem into individual hybrid portions and later fuses results of the clusters at the latter stages to overcome the curse of dimensionality. Figure 5.1 is the pictorial view of the HieFLANN_BP network. The procedures involved in the HieFLANN_BP are:

- Formation of homogenous feature spaces
- Hybrid learning
- Generalization learning

Figure 5.1 Pictorial view of the HieFLANN_BP network

5.2.1 Formation Of Homogenous Feature Space

As introduced in Chapter 4, the initial step to hierarchical construction is the identification of the Homogeneous Feature Spaces (HFS). The HFS is feature subset that contains similar features as compare to other heterogeneous feature subset. The central idea of creating the HFS is to facilitate learning process, such that similar features are

grouped together for the clustering process. Similar structures can also be found in the visual cortex and early visual pathways. Section 4.3.1 provides a detail explanation on how the HFS is formed.

5.2.2 Hybrid Learning

Before proceeding further, it is important to understand the modular level of a hybrid structure as this module will be used extensively across all modules within the hierarchical architecture. This section explains the single hybrid module containing a KFLANN and MLP-BP.

The single hybrid learning model combining the KFLANN with the MLP-BP is shown in Figure 5.2.

Figure 5.2 Pictorial view of the hybrid learning KFLANN and MLP-BP

As described in Chapter 3, the modified KFLANN is an unsupervised clustering algorithm that groups similar data patterns together producing consistent cluster formations regardless of the ordering on data pattern sequence by using the data pattern reshuffling property. This is a key fundamental requirement for cascading networks as it ensures consistency at the base layers of the network. The KFLANN can be viewed as vector quantization from the unlabeled data patterns, where clusters are intended as ‘natural structures’ or ‘statistical prototypes’ characterizing the dataset. Unfortunately, the clustered KFLANN does not possess generalization ability. This may hinder the objective of this work to develop an automated hierarchical learning system. Since learning systems

should address generalization issues, it is necessary to attach another mathematically well-founded function approximation framework to the KFLANN.

In this work, the Multilayer Perceptron Network (MLP) with classical error Backpropagation learning algorithm (MLP-BP) was used to form the hybrid learning model of KFLANN and BP. As such the notation for the network is given as KFLANN_BP. The main reason of choosing the MLP type network is its simplicity of topology and sound mathematical convergence in its learning theory [18], [72], [110]. The MLP-BP is well established as a universal approximator learns inductively from data exemplars. MLP type networks are able to generalize on any well-defined function and have the capability of interpolation after training. The selection of another function approximation model is possible as long as it is able to produce a Multiple Input Single Output (MISO) module.

From Chapter 3, the KFLANN, can also be classed as a Leader type algorithm [43] that is able to perform fast learning. Once the KFLANN is trained, it can then be used to train the interpolative Backpropagation MLP type network.

In psychological experiments, it has been found that subjects training to do a difficult task produced limited results on the first day. But after a day of lapse, subsequent trials show marked improvements from the time last training. This indicated that there was some form of continued learning even after the training ceased. From research by Hermann Ebbinghaus [47] the brain exhibits some kind of rehearsal patterns, which allow it to extend its memory, incorporating new memory exemplars with old. This rehearsal patterns continue immediately after the events, and also during sleep sessions. Through the use of functional Magnetic Resonance Imaging (fMRI) scans and EEG monitoring

devices, researchers have also found that the human sleep patterns exhibit systematic activations in different parts of the brain [83], [86]. Through these recorded evidence, the work here attempts to combine the fast learning component, KFLANN with the MLP-BP that require longer times to learn. In a very loose way, it models the paradigm of short term learning extending information into the longer term memories.

The proposed hybrid learning process can be viewed as self-supervised learning paradigm. The flowchart of the proposed hybrid learning model is given in Figure 5.3.

Figure 5.3 Flowchart of the hybrid learning model of KFLANN and MLP-BP

Hybrid learning begins with the initial KFLANN clustering of the raw data inputs. Since the KFLANN initial parameters can be obtained from statistical characteristics of the existing raw data, the clustering process can be fully automated. It eventually yields clusters that are used as target values for the supervised training of the MLP-BP. The clustering outcome from the KFLANN is therefore used to supervise the MLP-BP training process. The learning terminates when the MLP-BP converges to specified limits.

The input data is initially clustered by the KFLANN and each data pattern is allocated to a single class (cluster). Cluster encoding is the direct mapping of the cluster index. For example, a particular pattern may be processed to belong to cluster k . The cluster index k is encoded as the class label. At the end of the clustering process, all data points will be associated with one specific cluster where the cluster label may be an enumerated value. The MLP-BP is then trained using the cluster information where the associated cluster indices are known. The KFLANN clustering can be viewed as a process that prepares the target training value for MLP-BP training. Thus the first stage KFLANN provides the associative grouping and the second MLP stage provides the generalization power. Figure 5.4 depicts the structure of the hybrid learning model.

Figure 5.4 Structure of the hybrid learning model

As seen from Chapter 3, the KFLANN provides consistent clustering capabilities. As a result, the final outcome will be a consistent set of clusters with similar cluster centroids, as oppose to clustering algorithms that create different set of clusters with each different Data Presentation Sequence (DPS).

Within each HFS, a local distribution analysis takes place in each hybrid learning model of the HieFLANN_BP. Each individual HFS subnet operates independent from each other in such a way that each has its own unique set of tolerance and vigilance settings. It is easier to gain control over the granularity of the clustering in each hybrid subnet. The

hybrid learning subnet is less susceptible to interference since it functions on the weights relevant to the localized information.

A challenging issue in pattern classification tasks involves multiple heterogeneous features within a single domain. Conventional algorithms usually adopt a uniform scheme to handle the dataset. This is deemed to be inappropriate as it is difficult to address varying requirements of the datasets with a single generalized treatment. The HieFLANN_BP however takes this challenge well. It treats each heterogeneous feature subsets independently.

5.2.2.1 Performance Analysis Of Hybrid Learning Model KFLANN-BP

Experiments were conducted to investigate the efficiency of the hybrid learning model of KFLANN_BP using the UCI datasets. The typical KFLANN network initialization as described in Chapter 3 was performed. The MLP-BP weight connections were randomly initialized with value range between -1 and 1 . The clustering encoding from the KFLANN was used directly as the class label for the MLP-BP training to converge towards.

Dataset	Hybrid KFLANN_BP (%)	MLP-BP (%)	KFLANN (%)
Iris	96.97	100.0	90.48
Wine	96.67	92.86	90.77
Wisconsin Breast Cancer	93.95	94.00	91.77
Heart	80.00	78.33	77.14
Sonar	67.50	85.00	67.54

Table 5-1 Experimental results on hybrid learning model

The experimental results obtained were recorded in Table 5-1. The second, third and fourth columns records the accuracy measured (all accuracy mentioned in this chapter followed the *ROC* method unless otherwise mentioned) on for the hybrid KFLANN-BP, MLP-BP and KFLANN respectively. It was shown that in general hybridization of the

KFLANN and MLP-BP performed as well as the individual learning scheme. The overall accuracies obtained using the hybrid KFLANN-BP compared well with the individual KFLANN clustering algorithm and MLP-BP. The results showed that self-supervised learning hybrid model performed better than unsupervised learning algorithm.

Figure 5.5 (a) Difference between accuracy obtained by hybrid KFLANN-BP and MLP-BP (b) Difference between accuracy by hybrid KFLANN-BP and KFLANN

The differences between accuracies on the UCI datasets using the hybrid learning method, MLP-BP and KFLANN are showed in Figure 5.5, (a) is the accuracy obtained by KFLANN-BP minus MLP-BP's accuracy while (b) is the accuracy obtained by KFLANN-BP minus KFLANN's accuracy. It is noticed that self-supervised hybrid learning model's performance is comparable with the unsupervised KFLANN clustering algorithm.

Although MLP-BP performed slightly better than the hybrid learning method, the self-supervised hybrid learning method does not require class labels. The results obtained show comparable accuracy between the hybrid learning method and MLP-BP. The core objective of the thesis is to develop a self-learning hierarchical model. The emphasis is on the hybrid KFLANN-BP and how it compares with the KFLANN. A detailed explanation will be provided using Iris dataset as the dataset.

5.2.2.2 Hybrid Learning Model KFLANN-BP On Fisher Iris Flower Data Set

Fisher Iris flower dataset has the objective of classifying 3 groups of flowers according to their characteristics. The 3 classes of Iris flowers were Setosa, Virginica and Versicolor. Each class consists of 50 data instances with each form up of 4 attributes namely sepal length, sepal width, petal length and petal width. The tolerance value was obtained using the standard deviation method and the values are listed in Table 5-2. The experiment was conducted using a normalized dataset. The training set contained 120 data instances while the test set contained a random set of 30 data patterns.

Feature	Tolerance
Sepal width (F1)	0.1705
Sepal length (F2)	0.2298
Petal width (F3)	0.3177
Petal length (F4)	0.2952

Table 5-2 Feature tolerance for Iris data

The KFLANN clustering on Iris training formed 12 clusters. The second portion using the MLP-BP was subsequently trained with the identical training set and the KFLANN cluster assignment was used as the class label. Number of epoch for this second stage training was 8000. The accuracy was then subsequently obtained by comparing the cluster with the actual class label. For example from the 12 clusters of Iris data, cluster 1, 2 and 4 represent class Setosa, during the generalization testing, if a data pattern of class Setosa belong to any of the 1,2 or 4 cluster then it will be considered correctly classified.

Cluster	Class	Cluster	Class
1	Setosa	7	Versicolor
2	Setosa	8	Virginica
3	Versicolor	9	Versicolor
4	Setosa	10	Virginica
5	Virginica	11	Versicolor
6	Virginica	12	Versicolor

Table 5-3 Actual classes cluster mapping for Iris data

The MLP-BP residing within the hybrid learning model can be viewed as a distributed representation that protects the model from degradation. The gradient-descent method

adopted by BP allows the model to detect subtle changes during modeling. This phenomenon increases the fault tolerance ability of the KFLANN_BP model.

The KFLANN_BP hybrid learning model is a self-supervised model that requires no external supervision. The training cost is lower than the supervised MLP-BP because the automated training process is achieved by the first stage KFLANN. The hybrid learning model is very suitable for domains with tedious procedures to determine the actual target values. As a consequent, it is possible to perform on line training with this hybrid learning model. KFLANN teaching network with its stable cluster characteristic ensures that the supervision to the MLP-BP is consistent.

5.2.2.3 Comparisons Between the KFLANN_BP And KFLANN

Table 5-4 summaries the comparison between KFLANN_BP and KFLANN.

Features	KFLANN_BP	KFLANN
Learning Strategy	Self-supervised	Unsupervised
Topology	Combination of KFLANN and MLP network structure	Self-organized 2 layer network structure
Adaptability against overlap class distributions	Inherit the MLP-BP's feature in handling the overlapping class distribution	Difficult to cluster overlap regions properly
Formulation on Optimization Function	Yes	No

Table 5-4 Feature comparison between KFLANN-BP and BP

The advantages of the hybrid KFLANN_BP over the non hybrid counterpart (MLP-BP or KFLANN) are as follow

- The class label for the supervised BP training is provided by the KFLANN clustering algorithm.
- Single calibrated MLP-BP is sensitive to the consistency (accuracy) of the class label provided during the learning process. The problem is alleviated in the hybrid model.

- MLP-BP performs function optimization during the learning process that enables the hybrid network to derive a well defined objective function.
- The unsupervised KFLANN clustering algorithm is not a complete learning system since it does not attach with any adaptive learning. However the hybrid model further adapts the KFLANN output through the MLP-BP. This maps the input space of the hybrid network to the output space, similar to a function approximation solution space.

5.2.3 Generalization Learning

After completing the clustering process of the different HFSs by hybrid learning subnets, the hybrid KFLANN and MLP-BP learning component at the top most level is activated. It uses the information from the hybrid subnets, outputs from MLP-BP of all subnets that were fed forward as training data to the top most hybrid unit. The pictorial view is given in Figure 5.6 with S subnets.

Figure 5.6 Generalization learning in HieFLANN_BP with hybrid learning KFLANN and MLP-BP network at top most level

The MLP-BP network of each hybrid learning model is easier to train than if the whole problem was trained using a single MLP-BP. This explains why the hybrid network results encounter less training resistance than in single network. When comparing weight connections the hybrid learning subnet uses fewer number of weight connections while the non-modular networks require more.

5.3 PERFORMANCE ANALYSIS OF THE HIEFLANN_BP

Experimental testing was conducted on several sets of data to investigate the performance of HieFLANN_BP. All the KFLANNs in the hybrid learning subnets had vigilance settings at 0.9 and feature tolerances were calculated using the statistical standard deviation method. The input neurons integrated directly with the HFS. The number of input neurons equals to number of features in the given HFS. The formula to determine the number of hidden neuron is given in Appendix B. The MLP-BP network is the Multiple Input Single Output structure (MISO). With the MISO implementation, only single output neurons were used at the output layer for all hybrid subnets. All the MLP-BPs in the hybrid learning models resident in the hierarchical structure were single hidden layered networks.

5.3.1 Experimental Analysis Of HieFLANN_BP On UCI Datasets

The accuracy on each UCI dataset is shown in Table 5-5. The dataset description is found in the first column. The second and third columns are the accuracies of training and test sets respectively.

Data set	Accuracy (%)	
	Training	Test
Iris	93.53	100.0
Thyroid	88.70	94.90
Wisconsin Breast Cancer	95.52	97.35
Wine	100.0	100.0
Heart	85.19	83.33
Segment	80.06	80.66
Spambase	87.30	88.35
Sonar	94.64	80.00
Average	90.62	90.57

Table 5-5 Accuracy of HieFLANN_BP for UCI datasets

The accuracy obtained for test set is comparable with training set for most of the datasets.

This implies that the network was able to learn and generalize.

Figure 5.7 Training and testing accuracy for the HieFLANN_BP model

Figure 5.7 shows the histogram comparing the training and test data for 8 UCI datasets. A detailed explanation on how the HieFLANN_BP works using Wisconsin Breast Cancer dataset is given later.

5.3.2 HieFLANN_BP On The Wisconsin Breast Cancer Data Set

The Wisconsin Breast Cancer (WBC) data set is used to explain how the HieFLANN_BP performs classification. WBC consists of 9 numerical attributes and 2 classes. The detailed description of the dataset can be found in Appendix C.

5.3.2.1 Formation Of Homogenous Feature Space

The feature similarity matrix of the WBC dataset was created. The KFLANN clustering on the feature similarity data (as the one in Figure 5.8) produced the HFSs as in Table 5-6, first feature is represented by F1, second feature is denoted as F2, so on and so forth. There are 4 HFSs generated in this experiment testing. The first HFS had F1 while the second HFS space was characterized by F2, F3, F4, F7 and F8. The third HFS is represented by F6 and HFS4 was represented by F9.

HFS	Feature
1	F1
2	F2, F3, F4, F7, F8
3	F6
4	F9

Table 5-6 HFSs for WBC dataset

	F1	F2	F3	F4	F5	F6	F7	F8	F9
F1	0	6.9045	6.5276	8.0338	6.6266	7.1854	6.5228	7.8625	9.4818
F2	6.9045	0	3.4919	6.1007	5.3131	7.1905	5.3643	6.1134	8.259
F3	6.5276	3.4919	0	6.2833	5.405	6.905	5.3883	5.9466	8.2463
F4	8.0338	6.1007	6.2833	0	6.3342	7.6224	6.0596	7.1273	7.6365
F5	6.6266	5.3131	5.405	6.3342	0	7.5108	5.5626	6.3258	7.0254
F6	7.1854	7.1905	6.905	7.6224	7.5108	0	6.6941	8.2449	10.7104
F7	6.5228	5.3643	5.3883	6.0596	5.5626	6.6941	0	6.2389	8.2513
F8	7.8625	6.1134	5.9466	7.1273	6.3258	8.2449	6.2389	0	8.2039
F9	9.4818	8.259	8.2463	7.6365	7.0254	10.7104	8.2513	8.2039	0

Figure 5.8 Feature similarity for WBC dataset

After the HFSs creation, the HieFLANN_BP proceeded to the next step to generate the hybrid subnet according to the number of HFSs.

5.3.2.2 Hybrid Learning

As a consequence of the 4 HFSs created, there were 4 hybrid subnets of KFLANN-BP layering the HieFLANN_BP network as illustrated in Figure 5.9.

Figure 5.9 HieFLANN_BP network structure for WBC dataset

The KFLANN initially clustered the exemplars to form training classes for the later stage. Each hybrid subnet can be executed in parallel at this stage of hybrid learning MLP-BP module. All MLP-BP modules of hybrid subnets adopted the MISO network structure.

Hence the output of MLP-BP is a single dimensional vector. One advantage of this system is the strong definitions in modularity and scalability.

Any single subnet of the hierarchical network can be independently fine tuned on the specific characteristics of the problem to be solved. Thus the entire architecture can be easily modified if the localized problem description changes. By training only those units involved in the modification, the entire architecture can remain relevant. This is property is not illustrated in this thesis.

5.3.2.3 Generalization Learning

The last step in the HieFLANN_BP is the generalization learning using yet another hybrid KFLANN and MLP-BP network. The MLP-BP output collections from the 4 subnets are used as the input to the top hybrid module MLP-BP, Figure 5.10 depicts the scenario.

Figure 5.10 Pictorial view on the generalization learning of HieFLANN_BP for WBC dataset

After the convergence of the top most MLP-BP unit, the accuracy is computed to check the performance of the HieFLANN_BP on the WBC dataset.

5.3.3 Comparative Study of HieFLANN_BP with other Hierarchical Model

The proposed hierarchical neural network HieFLANN_BP was compared with some of the well established hierarchical neural network based model, there are Neocognitron [36], [37], Multi class Support Vector Machine Tree (m-SVMT) [76], [128]. The comparisons in this section focused on the characteristic of the network model itself and not the classification performance. Those hierarchical networks compared in this section shared a common property that is all adapt a divide and conquer strategy.

Support Vector Machine (SVM) is the processing unit form by the m-SVMT, the SVM employs structure risk minimisation principle. Generally, SVM is an optimal hyperplane algorithm that minimize a bound on the empirical error together with the complexity of the classifier concurrently.

Neocognitron is primarily designed for two dimensional pattern recognition hierarchical network [78], it has been mainly inspired by biological plausibility issues discovered in the vision neural system.

5.3.3.1 Network Construction

In the adaptive hierarchical multi-class SVM classifier, the network recursively partitions the current dataset into two disjoint subsets where there are used as positive and negative samples to train an SVM classifier. There will be $C-1$ SVM classifiers forming a hierarchical structure to discriminate C number classes of a given dataset. It is clear that this hierarchical multi-class SVM network adopts supervised method in its network construction and the number of classes has to be predefined. On the other hand, the HieFLANN_BP is data-driven in its network construction and no supervision required.

Neocognitron neural model was proposed by Fukushima [36], [37] from the inspiration of the visual nervous system. It is an unsupervised self-organized network. It consists of a cascade connection of a number of modular structures that made up of two layers of cells (denoted as S-cells and C-cells). Cell is a processing unit with characteristics analogous to a biological neuron with predefined activation function. The unsupervised self-

rganized method employed in the neocognitron network construction is similar with the HieFLANN_BP network. The processing unit cell is layering in plane structure involve weight updating each time presentation of a data pattern or stimuli. Processing unit of the HieFLANN_BP is the hybrid KFLANN_BP model, the KFLANN model updates the node information upon complete of a single epoch instead of every presentation of data pattern.

Figure 5.11 The hierarchical structure of the Neocognitron network, S is the S-cells in 2 dimensional plane, C represents C-cells layering in 2 dimensional plane, V is the V-cells situate in V-plane. The S-cell and C-cell need to have notation of cell type, layer where it is located, which cell plane is it and the location of the cell in the plane.

5.3.3.2 Network Parameter

There are two phase in the HieFLANN_BP network, the first phase known as the KFLANN clustering process. In this phase KFLANN network involves initialization of two parameters (tolerance and vigilance). In the neocognitron network there are several parameters need to be determined prior the network learning phase, there are selectivity, learning rate and three mask parameters. On top of these parameters attendant to neocognitron network, the supervised version of the neocognitron requires to specify the MLP network parameters such as momentum, number of hidden units, initial weights and activation function. There are consulation works providing guidelines to initial those MLP network parameters. However, in practical the best is to optimize the generalization performance by varying the network structure. The second phase of the hybrid unit KFLANN_BP involves the classical MLP with Backpropagation learning, it faces the

similar issues to obtain the best MLP structure to tackle the on hand task. The settles of optimal MLP network structure is not within the scope of this thesis.

5.3.3.3 Learning Paradigm

The HieFLANN_BP consists of two learning phases, the first phase involves the competitive learning and the second phase is a supervised Backpropagation learning. Generally learning paradigm for the HieFLANN_BP can be considered as self-supervised scheme. This is construct to the supervised learning use in the adaptive hierarchical multi-class SVM network.

The neocognitron network can learn either in unsupervised or supervised method depends on the objective of learning. This flexible scheme is advantagoues and can be adapted into HieFLANN_BP. When neocognitron network is used for classification purpose, it extracts and compress feature representations until the input is transformed into a vector. This vector contains elements of the similarity between the input pattern and classes that neocognitron has learned. Neocognitron adopts the winner take all scheme where highest output at the unit of the final layer determines the class assigned to the presented data pattern. This is similar to the fashion employed in the KFLANN cluster membership assignment.

5.3.4 Comparison Between HieFLANN_BP And HieFLANN

Table 5-7 is the experimental results obtained by the HieFLANN_BP and HieFLANN. The purpose is to compare the performance between the self-supervised hierarchical model HieFLANN and the HieFLANN_BP.

The network setting for HieFLANN_BP is same as the above while the HieFLANN network setting is identical with the setting described in Chapter 4 Section 4.4.

No	Data set	HieFLANN_BP (%)	HieFLANN (%)
1	Iris	100.0	95.24
2	Thyroid	94.90	95.00
3	Wisconsin Breast Cancer	97.35	97.23
4	Wine	100.0	96.67
5	Heart	83.33	87.53
6	Segment	80.66	82.26
7	Spambase	88.35	83.14
8	Sonar	80.00	75.19

Table 5-7 Comparison on accuracy of HieFLANN_BP with hierarchical model HieFLANN

The average of accuracies obtained using the HieFLANN_BP was 90.57% and 89.03% for HieFLANN. Although this difference is small, the interpolation ability exhibited in the HieFLANN_BP does not exist in the HieFLANN. This is because the C2FeCom feeds information to another layer of KFLANN that cannot perform function approximation. Furthermore, the computational complexity of HieFLANN_BP is less as compared to the HieFLANN since it does not involve the eigen-analysis.

Figure 5.12 (a) Difference in accuracy of HieFLANN_BP and HieFLANN, (b) In the standard deviation unit

Figure 5.12 is the plot to highlight the difference in accuracy between the HieFLANN_BP and HieFLANN. The numbering of the dataset is consistent with Table 5-7. There were 5 datasets in which HieFLANN_BP performed better than HieFLANN. These were the Iris, WBC, Wine, Spambase and Sonar datasets. For Thyroid, Heart and Segment dataset, the accuracies are comparable for both HieFLANN_BP and HieFLANN. The difference in accuracy for Thyroid and Segment dataset was insignificant.

5.3.5 Comparison Between HieFLANN_BP With Non Hierarchical Network KFLANN And MLP-BP

The comparison of the prediction accuracy with non hierarchical model is given in Table 5-8. MLP-BP and KFLANN is single calibrated algorithms that take in the entire feature space.

No	Data set	HieFLANN_BP (%)	MLP-BP (%)	KFLANN (%)
1	Iris	100.0	100.0	100.0
2	Thyroid	94.90	85.33	92.00
3	WBC	97.35	94.00	91.77
4	Wine	100.0	92.86	90.77
5	Heart	83.33	78.33	77.14
6	Segment	80.66	63.04	88.43
7	Spambase	88.35	83.87	86.01
8	Sonar	80.00	85.00	67.54

Table 5-8 Accuracy comparison of HieFLANN_BP with non hierarchical model

The supervised MLP-BP and unsupervised KFLANN were used to investigate how the hierarchical based model performs over the non-hierarchical network. The results show that the HieFLANN_BP outperformed the non-hierarchical (flat) KFLANN and MLP-BP.

Figure 5.13 (a) Difference in accuracy of HieFLANN_BP and MLP-BP and (b) In the standard deviation unit

The plot in Figure 5.13 is the difference between the accuracy obtained by HieFLANN_BP and MLP-BP. Vertical axis is the difference and the horizontal axis is the dataset index according to Table 5-8. In 6 out of the 8 datasets tested, the HieFLANN_BP performed better than the MLP-BP.

Figure 5.14 (a) Difference in accuracy of HieFLANN_BP and KFLANN, (b) In the standard deviation unit

The HieFLANN_BP has accuracy higher than the KFLANN for all datasets except Segment dataset. From the statistical point of view, the difference in the accuracy measurement between the HieFLANN_BP and the KFLANN is significant since the difference between accuracy in standard deviation unit as shown in Figure 5.14 (b), most of them are at least a standard deviation away.

5.4 CHAPTER SUMMARY

This chapter presented a dedicated pattern recognition (classification) technique using hierarchical structural hybrid learning model HieFLANN_BP. The HieFLANN_BP inherits the stability resident in the KFLANN algorithm to provide a consistent classification output. The DPS does not cause inconsistent clustering outputs. As a result, a similar clustering output was obtained as long as the complete dataset was used. Experimental evaluation on the hybrid learning model KFLANN_BP showed that the self-supervised hybrid learning performed better than its single calibrated unsupervised KFLANN. A comparison of the characteristic of the KFLANN_BP and KFLANN highlights the benefits of the hybrid learning over the unsupervised KFLANN. The most significant advantage is that the KFLANN_BP displays the interpolation ability with well-defined objective functions. This is difficult to achieve in the standalone KFLANN.

Unlike the HieFLANN, where statistical quantities and computationally intensive matrix calculations are needed, the HieFLANN_BP model is less complex. The results documented in this chapter illustrate the capabilities of the hybrid learning model to solve pattern recognition problems of large dimensionality domains.

In order to tackle curse of dimensionality the hybrid learning model is used as an atomic component. And complex problem is divided into several smaller problems. Sub-problems are then solved using existing algorithms. The once impossible task of resolving large dimensionalities can now be achieved. This modular approach was implemented to solve various problems.

By using accuracy and known ground-truth classifications, the efficacy of the proposed model was tested on a wide range of domains. It is important to note that the hierarchical classification corresponding to optimal hybrid methodology delivered an excellent performance. This evidence was obtained from the experimental results shown in Section 5.3.

As in the Section 5.3.3, the comparison between HieFLANN_BP and HieFLANN showed that HieFLANN_BP and HieFLANN both had comparable performances. The HieFLANN_BP however has the interpolation ability embedded in its MLP-BP unit while the HieFLANN does not exhibit this property. A further observation of both algorithms indicates that the computational complexity of HieFLANN_BP is less when compared to the HieFLANN.

There are similarities and differences when performed the characteristic comparison between the HieFLANN_BP with other hierarchical neural networks. We cannot

comment that the differences in model's features entitle which network model is better than the other. In practical points of view, HieFLANN_BP can be considered less complexity compared to the neocognitron and m-SVMT. The atomic processing unit of the neocognitron needs to tackle a number of parameters due to its connections and each connection involves computation during for every presentation of data pattern. Parallelism can be implemented in the HieFLANN_BP without much effort since each unit work mutual exclusively. However in the neocognitron and m-SVMT, additional efforts needed to be put in for parallel processing. In [13], [27] and [128], the parallel implementation of SVM has been proposed. In [27], parallel implementation of the SVM has to take into account of some issues including kernel caching, computation of kernel matrix, selection of the working set, shrinking strategy and stopping criterion. It is clear that strategy with higher complexity is required to implement the parallel SVM. Parallel method proposed in [13] for SVM considers only one class by one class. The parallel strategy employed in [128] need to combine sequential minimal optimization, SVM decision tree and cluster.

The HieFLANN_BP has many advantages over its non-modular (non hierarchical) counterparts. In Section 5.3.4, the comparison on the experimental results between the HieFLANN_BP and its non-hierarchical model KFLANN and MLP-BP provides evidence of those advantages. The HieFLANN_BP model imposes simple modular learning through the established hybrid subnets. It is easier to learn simple functions independently rather than to learn a complex function within a single large network. Each hybrid learning model residing within the HieFLANN_BP network is simpler than the entire flat non-modular network solving the same problem. From the results obtained, it is fair to say that the HieFLANN_BP provides better generalization ability than the non-hierarchical networks.

6 FEATURE SELECTION

This chapter discusses the need for feature selection within the framework of hierarchical networks. Despite the efficacy of the feature compression method mentioned in Chapter 4 and the HieFLANN_BP in Chapter 5, there is still a need to ensure that the hierarchical structures are formed with relevant feature samples that bear significant correlation value to the output. Neural network computations still possess the fundamental premise of Garbage-in, Garbage-out.

In order to provide flexibility of the proposed frameworks of hierarchical neural networks to resolve complex problem domains, it is necessary to adopt measures to seek relevant features as inputs. This work proposes a selection based on feature PUrity and RElevancy (PURE).

6.1 IMPORTANCE OF FEATURE SELECTION

Feature selection is a crucial pre-processing step to facilitate the learning process through ‘optimal’ feature subset selection so that only necessary features are used. The focus of this chapter is on feature purity and feature relevance analysis. While cascading networks are important in resolving the curse of dimensionality and implementing K-Set type structures, it would be difficult if networks were plagued with irrelevant information. A supervised scheme is introduced, providing ranking to features base on their PUrity and RElevancy. This process does not devoid the unsupervised building of hierarchical neural networks because it is placed as a pre-processing stage. It essentially rids the feature space of useless features. Purity is the measure of uniqueness of features amongst classes. Relevance is a relationship between a feature and various individual classes and its

relationship with other features. This work suggests that when feature purity is combined with relevance, a resultant feature evaluation index, denoted as PURE (PUrity and RElevancy) is formed.

The HieFLANN and HieFLANN_BP proposed in this thesis performed well when handling large dimensionality domains, however if domains contaminated with vast amount of irrelevant (noisy, redundant) attributes were used, the performance of the network will degrade. Feature selection is therefore useful to filter irrelevant attributes and it facilitates the pattern classification task.

To illustrate the effects of useless feature, the Fisher Iris data was used. In the Fisher Iris data, three types of Iris flowers namely the Setosa, Virginica and Versicolor form the dataset and each category is described by four attributes. These four attributes are sepal width (F1), sepal length (F2), petal width (F3) and petal length (F4). It is known that the Virginica class and the Versicolor class are highly similar in the sepal length and sepal width dimensions. The Setosa class however is well discriminated over the range of numerical values for the 4 attributes from the other 2 classes.

Figure 6.1 3D plot of Iris features

The 3D plot of Iris flower according to their flower types in Figure 6.1 shows that feature group (F1, F2, F3) and (F1, F2, F4) are not suitable for discriminating analysis due to the inseparability of the data. Without specifically knowing the representations, it can be seen that only feature groups (F1, F3, F4) and (F2, F3, F4) provide better options for classification purpose, separating the three types of representations. This example illustrates that relevant feature selection is important, filtering irrelevant features to aid the classification task.

Classifier systems respond adversely with irrelevant (noise, redundant) features. Due to irrelevant features residing in features used for analysis, any classifier system will be detracted. Classifier systems operating with relevant feature subsets will perform better than those that included irrelevant features within the feature space. As a consequence of feature selection, the processing time can be reduced and memory storage is minimized due to fewer features being processed.

The curse of dimensionality caused by large dimensionality indirectly encourages the need to use feature selection prior to classification process. Other than from the benefits previously mentioned, feature selection impacts the pattern classification process.

- (a) Reduce the number of instances that are needed for learning a sufficiently accurate classification,
- (b) Reduce the task of data collection (only salient features are collected instead of the huge chunk of data).

Ben Bassat [9] classified the feature selection methods based on different evaluation indices such as information (uncertainty measure), distance measurement and dependency computation. In 1992, Doak [26] also grouped the feature selection methods according to the feature evaluation index namely data intrinsic (distance, entropy, dependence and etc), classification error and estimated error. Langley [11] categorized the feature selection into 2 groups, namely the filter and wrapper based approach. Dash and Liu [22] summarized the grouping correspond to the 5 feature evaluation indexes and the 3 searching schemes yielded 15 possible types of feature selection combination. With the advance of computer technology, the emergence of large size feature dimensionality datasets warrant the need for better machine learning methodologies. These, new problem definitions demanded feature selection schemes that could keep in pace with the requirements.

Despite the wide range of feature selection methodologies available, this chapter focuses on two groups of feature selection methods, namely the filter and wrapper. The filter method selects features according to the defined feature evaluation index while the wrapper approach relies on the learning algorithm to extract features. The filter based feature selection approach requires an efficient feature ‘goodness’ measurement. Many measurement indices have already been developed. Feature evaluation index plays important role in determining the efficacy of the feature selection method. For example, only using the feature-class relevancy is insufficient to retrieve good feature subset. The

Entropy index that is used to select good features tend to favor those features with more values [70] and [122]. This may cause biases since features with more variations may not be good features. Consistency [89], [90] is a common feature evaluation index used in such a way that a good feature should display consistent characteristic in presence of different classes. However too much emphasis on the consistency may lead to an over fit of the problem. The wrapper method was also shown to be useful but was slow due to involvement of learning algorithm. It is difficult to implement wrapper based feature selection algorithms for large dimensional datasets. Filter based feature selection algorithms tend to bear significant advantages as it functions independently of the learning algorithm. Some references about feature selections methods were discussed in [35], [94], [102].

6.2 FEATURE PURITY

Feature purity is an index that measures the goodness of a feature to represent a class and this goodness is relative measure that is compared to other classes. A good feature for pattern classification should have minimal overlaps with other classes. Feature purity measurement index, Pu for feature X proposed in this thesis is formularized as:

$$Pu(X) = 1 - \frac{Ov(X)}{nC} \quad (6-1)$$

$$Ov(X) = \sum_{j=1}^C \sum_{k=1, j \neq k}^C \left| \left\{ m_x \mid X^{C_j} \cap X^{C_k} \neq \{ \}, j \neq k \right\} \right| \quad (6-2)$$

Where C is the number of classes, $Ov(X)$ is the overlapping feature, X^{C_j} is the instances that belong to class j , m_x is the set of X that is overlap in both class j and class i .

Pu has the properties:

- (a) $Pu(X) = 1$ if and only if feature X is non overlap in representing classes.
- (b) $Pu(X) = 0$ when feature X is highly overlap amongst classes.
- (c) $0 \leq Pu(X) \leq 1$

The feature purity measurement index can be adopted as feature goodness for classification as it measures the effectiveness of the feature in discriminating amongst classes. Features bearing higher purity values are more likely to be selected, compared to the ones with lower purity values. The index does not bias to any statistical inference, thus it does not cause undesirable negative effects.

Feature purity does not bias to any statistical interference. It is independently computed based on individual features. Thus it may not be possible to capture the interaction amongst features. This leads to the feature evaluation index hybridization of purity and relevancy. The proposed method provides a measurement of how good each feature discriminates amongst classes. It then becomes an effective feature ranking method.

6.3 FEATURE RELEVANCE

Feature relevance is the measurement on the correlation between a particular feature and the member of class also the relation with other features. A good feature is the one that is highly relevant to class but independent from other features. A good feature should therefore display the following characteristics:

- The feature should be independent of other features. Relevance between the particular feature and other features does not fall into the level that can be predicted by any of the other feature.
- The feature should have a strong relevance with the class so that there is sufficient confidence to predict the class. (High relevance with classes)

6.3.1 Feature Relevance for Continuous Feature Types

The commonly used method for relevance measurement of continuous feature type is the correlation coefficient. There are two methods to measure the association between

variables. One is the Pearson correlation coefficient [31] and the other is based on the information theory. The correlation coefficient used in this work computes the linear association between two variables. The formulation of correlation coefficient, for a pair of variable (X, Y) is as follow [31]:

$$CorrCoef(X, Y) = \frac{\sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}{\left(\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)^2} \right) \left(\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \mu_y)^2} \right)} \quad (6-3)$$

The numerator is the covariance while the denominator is the standard deviation. The characteristics of the correlation coefficient are summarized here [31].

- (a) The value of correlation coefficient range between -1 and 1 , this bounded range eases the interpretation on relevance between variables measured.
- (b) A pair of variables is highly correlated if their *CorrCoef* value is close to unity. If the *CorrCoef* value is near zero, then the pair of variables are said to be independent. This helps to detect the redundant features. With this property, it removes features that are near zero correlation to the class since they don't contribute to the class discrimination analysis.
- (c) Symmetry is a desired property in measuring the correlation between variables.

The feature relevance method has some limitations when used in the feature selection. The *CorrCoef* is only capable of capturing the linear association between a pair of variables but fails to detect nonlinear correlations.

6.3.2 Feature Relevancy for Nominal Feature Types

A nominal feature type consists of Boolean values 0 or 1. The relevance between the features can be measured using the Phi Coefficient (ϕ) that originated from [129]. The computation involves the forming of a contingency table and its computation equation is given below.

$$\varphi = \frac{ad - bc}{\sqrt{efgh}}$$

	X_i+	X_i-	Total
X_j+	a	b	e
X_j-	c	d	f
Total	g	h	n

X_i+ is the Boolean value 1 of i^{th} feature X_i- is the Boolean value 0 of i^{th} feature.

The Phi coefficient measures the degree of relevance between two binary variables. It has similar ways of interpretation as in the Pearson correlation coefficient. The denominator of the Phi Coefficient is normalized and this ensures the value falls within the range of -1 and 1.

6.4 THE PROPOSED FEATURE EVALUATION INDEX: PURITY AND RELEVANCY

As seen in the previous section, several existing metrics are already available. The attempt is to combine these to form a more comprehensive tool to measure Purity and Relevance.

6.4.1 Methodology for PURE Feature Selection

A good feature should exhibit significant relevance to classes yet be independent of other features. A good feature must also exhibit minimal overlapping amongst classes. This means that the mathematical formulation to obtain good features requires the maximization of the feature-classes relevance and purity while minimizing the inter feature relevance. This leads to the formalization of PURE feature evaluation index on feature X_i which is given as:

$$PURE(X_i) = \frac{\sum_{j=1}^C |CorrCoef(X_i, Class_j)|}{\frac{1}{d-1} \sum_{j=1, i \neq j}^d |CorrCoef(X_i, X_j)|} + Pu(X_i) \quad (6-4)$$

The first term of the formulation is the relevance measurement which involves the inter-feature correlations and feature-classes correlation. In general interpretations on the relevancy measure are

(a) $0.7 \leq |CorrCoef(X,Y)| \leq 1.0$ is considered strong relevance.

(b) $|CorrCoef(X,Y)| \leq 0.2$ is considered weak relevance.

The second term is the feature purity. The proposed feature evaluation index, PURE was implemented in the simple feature selection algorithm as described in Figure 6.2.

```

Input: ( $X_1, X_2, \dots, X_d$ ) // entire feature space
IrrF = {} //irrelevant feature
FS = {} //feature selected
TFS = {} //temporary feature selected
1 For  $i=1$  to total number of features
  Calculate  $Pu(X_i)$ ,  $CorrCoef(X_i, \text{Classes})$ 
  If  $Pu(X_i) > 0.7$ 
     $TFS \cup X_i$ 
  Else If  $CorrCoef(X_i, \text{Classes}) < 0.2$ 
     $IrrF \cup X_i$ 
  Else
     $TFS \cup X_i$ 
  End If
End For
2 Calculate  $PURE(X_i)$  For  $\forall i \in TFS$ 
 $FS = PURE(X_i)$  For  $\forall i \in TFS$ 
3  $FS = \text{Sort } FS \text{ in descending order}$ 
4 Select feature subset base on

$$\frac{\sum_{i=1}^{d_s'} FS(i)}{\sum_{i=1}^{d_s} FS(i)} \geq \tau$$

Output: reduce feature space
Notation:
 $d_s$  is the number of features that are temporary
selected.  $d_s'$  is the number of feature that are
selected.

```

Figure 6.2 Algorithm for PURE feature selection

The feature selection process comprises of 4 simple steps. The initial step is the purity computation on the entire feature space. In this step, the feature purity is calculated before the feature-class relevancy computation to filter out irrelevant features. If features exhibit purity values close to unity, it implies a heavy dominance. Dominant features can be

distinguished as independent classes with little or no overlapping. These features are likely to be selected as they dominate and would probably discriminate the classes effectively. If there are no features bearing purity value greater than 0.7, then it does not mean that no features were selected. The algorithm proceeds to check for the feature-class relevancy. When the feature-class relevance value is greater than 0.2, the particular feature will be temporary selected for the next step of feature filtering. For the purpose of minimizing the computational burden, the proposed feature selection algorithm will be instructed to include dominant features and exclude features having feature-class relevancy of less than 0.2.

This is followed by the PURE value computation for all elements that are temporarily selected features in the previous step. Then the algorithm proceeds to sorting the temporary filtered feature list in descending order.

The last process involves the feature selection process where the feature with PURE values greater than the threshold value are selected using the criteria of $0.5 < \tau < 0.8$ and τ is the threshold parameter defined. The default setting ranges from 0.5 to 0.8 and the information from domain expert is the most preferable.

6.4.2 Comparative Study of PURE Feature Selection With Other Feature Selection Methods

Feature extraction reduces feature dimensionality in the way different from the feature selection. In feature selection, best subset of input feature was chosen based on criterion function defined. While in feature extraction, the original feature space is transformed into reduced feature space (a new feature space is created). Linear Discrimination

Analysis (LDA) [120] and Principal Discrimination Analysis (PCA) [58], [131] adopts the method of transforming original feature space onto reduced dimensional space, both had numerous enhancement editions since past decade. In depth study of LDA and PCA is not in the scope of this thesis, instead the basic introduction on these methods was illustrated and comparison made with PURE feature selection. Summaries about LDA, PCA and PURE are given in Table 6-1.

Feature	LDA	PCA	PURE
Objective	Optimize the largest ratio of between class variation and within class variation when projecting the original feature to a subspace.	Optimize the transformation matrix by finding the largest variations in the original feature space.	Maximize the index of between classes features correlation and minimize the between feature correlation.
Assumption	No assumption needed.	Most information about classes is contained in the direction along which the variations are the largest.	No assumption needed.
Regulation	The reduced dimensionality is upper bound by $c-1$, where c is the number of classes.	The reduced dimensionality is upper bound by $n-c$, where n is total number of data patterns and c is the number of classes.	No stringent regulation applied.

Table 6-1 Comparison Between LDA, PCA and PURE

Many modified approach to LDA and PCA have been proposed, long lists of references include [62], [95], [96], [97], [113], [119]. Optimization criteria of PCA are different from the classifier's minimum classification error criterion. This may cause inconsistency between feature selection and the classification [81]. Recent enhancement for PCA and LDA is the online version to support dynamic learning environment. The proposed PURE feature selection method implements batch learning. For future, incorporate dynamic feature selection in the PURE method to accommodate the requirements of real life problem sets is a necessity.

Entropy based [33], Chi Square statistics [75], correlation based feature selection [42] methods are similar to the scheme applied in the PURE where ranking index is computed over the feature space and low rank features are filtered out. Basic introduction of the three groups of these feature selection methods are described in this section for comparison of their characteristics.

Entropy based feature selection method filters out features that are high in the entropy value and consider those with lowest entropy values. Entropy based method implements the minimum description length (MDL) [105] principle states that the best way to infer from training data is the one that minimizes the complexity of the theory and the length of the data encoded with respect to the theory. One problem encountered in this method is that it is possible to have well train model with zero entropy value (predict the data perfectly) but in fact it is not a good model. This can be explained by considering a feature X that partition the data, there will be exactly one value Y (with probability 1) in each of partition. Entropy value of Y with respect to X will be zero. The model unlikely to generalize well to unseen data, this is overfitting phenomena.

Chi Squared method uses chi squared statistic with respect to the classes to compute chi square value for each feature individually. This is followed by sorting the features chi square values in descend mode and features with larger chi square values are selected. Discretization pre-processing is needed when handling numerical features. Discretization process introduces variance to the feature selection process since different discretization schemes adopt different parameter yields variance in the output. Results of feature ranking depend on discretization procedure for continuous features, determine which discretization method to best fit the problem set is a challenging task.

Correlation based feature selection is the closest to the proposed PURE scheme. The difference between correlation based feature with PURE is that rather than ranking individual feature the correlation based feature selection method ranks subset of features. Heuristic best first search method is always used to facilitate the feature subset selection. Best first search beliefs that good feature subsets contain features highly correlated with class but uncorrelated with other features. The PURE feature selection method also adopts this concept. The disadvantage of the correlation based method is due to best first search method employed, different starting search point may end at different end state. The inconsistent of the feature subsets selection should be avoided. Exhaustive enumeration of all possible feature subsets is a limitation of this method.

6.5 PERFORMANCE ANALYSIS OF PURE FEATURE SELECTION METHOD

Experiment evaluation was carried out using three different categories of datasets:

- Dataset contains irrelevant, redundant and noise features.
- Dataset with number of instances far smaller than feature dimensionality, ($n \ll d$).
- Dataset with large feature dimensionality, the feature dimensionality is far smaller than number of instances ($d \ll n$).

6.5.1 Experimental Setup

The proposed PURE feature selection is a supervised method. For the ease of evaluation a simple supervised classification algorithm C4.5 was used. The experiment conducted was to compare performance of C4.5 learning algorithm on original feature space and reduced feature space after feature selection. Feature selection was carried out to filter the training set only. This is followed by the preparation of test sets using the selected feature(s). The

accuracy was measured on the test set using the *ROC* measurement (Appendix B). In other words, training set is used in feature selections and test set is used for evaluation purpose.

To perform the benchmark experiment, the Waikato Environment for Knowledge Analysis (Weka) [52] implementation of C4.5 was used. Weka is an open source Java based machine learning platform that provides a collection of learning algorithms under a user friendly interface. C4.5 is a supervised driven classification method, it starts with dataset belonging to known classes and is capable to deal with any mixture of nominal and numeric properties describe the data patterns. C4.5 algorithm scrutinizes on the data patterns to provide class discrimination. The induced information is then expressed in the form of decision trees or sets of if-then rules. A detail description of C4.5 algorithm on its advantageous and limitations can be found in [100].

There was no need for prior knowledge of the datasets. Two criteria were adopted in this work

- (a) Accuracy on the selected features (according to *ROC* method in Appendix B).
- (b) Number of selected features as a means to measure the effectiveness of feature selection in preserving the distribution of the original feature space.

The original feature space was first presented to C4.5 algorithm to obtain the accuracy without filtering. This was followed by the filtered feature space that was used and the accuracy was recorded. The comparisons were made between original feature space and reduced feature space in term of accuracy.

6.5.2 Experimental Results of PURE Algorithm on UCI Datasets

Table 6-2 shows the experimental results for UCI datasets. The indirect evaluation was necessary to check a classification algorithm's performance before and after feature selection. The second column records the accuracy for the original feature space, followed by the accuracy on the selected feature subset space. Results in Table 6-2 showed that the performance of C4.5 improved in general. The accuracy is higher when feature filtering is done. The average accuracy non-filtered datasets (continuous feature typed) is 80.55% while the PURE feature selection method obtained average accuracy 82.55%. R values for all datasets were close to unity, implying that the feature selection preserved the class discriminative power on the filtered feature space. The last column is the number of the selected feature (SF).

Dataset	Entire Space (%)	PURE (%)	R	d	SF
Segment	91.08	89.25	0.9835	19	7
Ionosphere	77.63	85.32	1.0671	34	16
Dermatology	96.37	93.99	0.9775	34	11
Waveform	77.31	79.04	1.0143	21	14
Spambase	85.37	89.35	1.0359	57	38
Sonar	70.43	70.95	1.0040	208	11
Arrhythmia	52.91	59.36	1.0458	279	82
Multi Feature	93.33	93.13	1.0018	649	126

Table 6-2 Accuracy for continuous feature type UCI datasets using PURE feature selection method

The plot on the difference between the accuracy obtained on original feature space and filtered feature space (accuracy of filtered feature space minus the one of the original space) in Figure 6.3 indicated that 5 out of the 8 datasets using PURE was able to improve the performance of the C4.5 classification algorithm.

Figure 6.3 Difference in accuracy between reduced feature space using PURE feature selection method and the original feature space for continuous feature type UCI datasets

Three feature evaluation indexes namely Correlation Feature Selection [42], Consistency [89], [90] and ReliefF [107] were used for comparison purpose. Correlation Feature Selection method evaluates the goodness of a feature subset by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that were highly correlated with the class while having low correlation with other features were selected. The Consistency approach examines the worth of a feature subset by the level of consistency in the class values when the training instances were projected onto the subset of attributes. The Consistency is used for comparison because it attempts to remove inconsistent features, which is synonymous to Purity which extracts pure feature representing the classes with minimum overlapping. ReliefF is a supervised scheme that used the distance measure in the feature selection process. All three feature selection methods were implemented using the Weka platform.

Dataset	PURE (%)	CFS (%)	SF	Consistency (%)	SF	ReliefF (%)	SF
Segment	89.25	90.00	6	91.51	14	88.96	8
Ionosphere	85.32	81.47	12	86.75	8	87.25	7
Dermatology	93.99	94.72	13	86.38	9	65.43	12
Waveform	79.04	77.64	15	75.03	13	74.93	5
Spambase	89.35	91.50	12	92.15	22	89.90	10
Sonar	70.95	75.19	9	72.80	12	66.75	5
Arrhythmia	59.36	55.25	13	35.66	18	31.55	12
Multi Feature	93.13	94.63	90	91.87	7	84.35	17

Table 6-3 Accuracy for continuous feature type dataset using correlation feature selection (CFS), Consistency and ReliefF feature selection method

On the average, the Correlation Feature Selection method achieved 82.55% while Consistency obtained 79.02%, ReliefF 73.64%. The PURE approach obtained 82.55%. Comparing to the other three methods, PURE method performed well on the tested datasets.

For the Segment dataset, all feature selection methods can be said to have comparable results since their differences are small (less than 3%). By using PURE method, Ionosphere dataset obtained higher accuracy than Correlation Feature Selection method, the difference was 3.85%. Although the Consistency and ReliefF method achieved better accuracy compared to PURE method but the differences are small (1.43% and 1.93% respectively). PURE, Consistency and ReliefF methods were comparable for this dataset. PURE method works better on the Dermatology dataset than the Consistency and ReliefF methods. For Waveform and Arrhythmia datasets, PURE method was the best performance for all the methods. By using the Consistency method, the highest accuracy was obtained for the Spambase dataset, PURE method was less efficient when compared to the rest of the methods for this dataset. PURE method produced higher accuracy for Sonar dataset than the ReliefF method, but not when compared to Correlation Feature Selection and Consistency method. For Multi-feature dataset, PURE is the second best method after Correlation Feature Selection method.

The discrete feature type datasets were also tested, the accuracy were recorded in Table 6-4.

Dataset	Entire Space (%)	PURE (%)	R	d	SF
Monks3	88.33	88.33	1.0000	17	2
Soya bean	82.37	87.84	1.0487	47	39
Splice	89.80	91.00	1.0110	60	7
Zoo	88.10	92.86	1.0444	17	9

Table 6-4 Accuracy for discrete feature type of UCI dataset using PURE feature selection method

The results show that the R values were all at least 1, implying that the filtered feature space was able to maintain the discriminative power of the original feature space.

The results show that the accuracies obtained when using the reduced feature space are better than the one with original feature space. The PURE feature selection method improves the performance of the C4.5 for all tested discrete feature type datasets. Further investigation on the proposed PURE feature selection method, the comparison with other feature selection methods was performed and the results were given in Table 6-5.

Dataset	CFS (%)	SF	Consistency (%)	SF	ReliefF (%)	SF
Monks3	88.33	2	88.35	2	88.35	4
Soya bean	76.43	39	83.34	17	76.86	36
Splice	91.97	6	95.00	11	91.13	6
Zoo	92.86	9	85.71	5	92.85	9

Table 6-5 Accuracy for discrete feature type dataset using CFS, consistency and ReliefF feature selection method

Table 6-6 show the difference in accuracy between different feature selection methods:

PURE, Correlation Feature Selection, Consistency and ReliefF.

Dataset	PURE - CFS (%)	PURE – Consistency (%)	PURE – ReliefF (%)
Monks3	0	-0.02	-0.02
Soya bean	11.41	4.5	10.98
Splice	-0.97	-4	-0.13
Zoo	0	7.15	0.01

Table 6-6 Difference in accuracy for discrete feature type dataset using PURE, correlation feature selection (CFS), consistency and ReliefF feature selection method

For Monk3 dataset, all four feature selection methods include PURE are comparable in their feature selection ability, the difference between the accuracy was small. PURE method worked well in feature selection for Soya bean dataset compared to other methods. Correlation Feature Selection and Consistency methods work better in feature selection for Splice dataset compared to PURE and ReliefF methods. The difference of the accuracies for Zoo dataset when using PURE, Correlation Feature Selection and ReliefF method were insignificant, however Consistency method seem to be less effective for this dataset.

In general, for the continuous and discrete type feature domains tested above, PURE feature selection method seemed to perform the best or comparable with the other three methods.

6.5.3 Experimental Results of PURE Algorithm on Micorarray Datasets

Microarray dataset is a real life dataset that does not clearly define the relevant features. The feature relevance can only be determined through the learning algorithm. Details information is given in Appendix C. The experimental evaluation on Microarray datasets using PURE is shown in Table 6-7.

Dataset	Entire Space (%)	PURE (%)	R	d	SF
Leukemia (aml, all)	91.45	91.45	1.0000	24481	1
Colon Tumor	58.89	80.00	1.1759	20000	32
Lung cancer	58.45	88.50	1.2695	12553	16
Ovarian Cancer	95.55	95.55	1.0000	15154	6
Prostate Cancer	54.45	100.00	1.4555	12600	1
Average	71.75	91.10	-	-	-

Table 6-7 Accuracy using PURE feature selection method for Microarray datasets

The PURE feature selection method awards significance to features selected in this domain. This is indicated by the reduced feature space, yet high accuracy. The improvement brought by the PURE feature selection method on the accuracy is significant. Furthermore the R values of all datasets of these domains were at least 1. This means that the reduced feature space was able to preserve the information in the original feature space from the perspective of class discriminations.

6.5.4 Experimental Results Of PURE Algorithm On NIPS 2003

Datasets

Neural Information Processing Symposium (NIPS) 2003 datasets were used for the feature selection competition in the symposium, detail information is described in Appendix C. Accuracies obtained for the NIPS 2003 datasets were summarized in Table 6-8. The results for PURE feature selection method are comparable to the results on the entire feature space (73.98% versus 73.12%).

Dataset	Entire Space (%)	d	PURE (%)	R	SF	CFS (%)	SF
Madelon	62.75	500	63.15	1.0029	11	58.95	39
Gisette	94.20	5000	89.00	0.9530	21	93.10	73
Arcane	65.00	10000	67.20	1.0165	11	58.75	39
Average	73.98		73.12	-	-	70.27	-

Table 6-8 Accuracy using PURE and correlation feature selection (CFS) method for NIPS 2003 datasets

Table 6-9 displays the difference between accuracy for the PURE feature selection method and Correlation Feature Selection method, also the comparison of the accuracy obtained using C4.5 algorithm for entire feature space and reduced feature space.

Dataset	PURE - Entire Space (%)	PURE – CFS (%)
Madelon	0.40	4.20
Gisette	-5.20	-4.10
Arcane	2.20	8.45

Table 6-9 Difference in accuracy obtained using PURE and correlation feature selection (CFS) method for NIPS 2003 datasets

It is noticed that Arcane and Madelon datasets had better accuracy in the reduced feature space compared to original feature space.

Results obtained for three categories of datasets, UCI, Microarray and NIPS 2003 datasets showed that reduced feature space eliminate the detrimental effect of irrelevant features. Considerable work was performed on high dimensional data. Some studies were done on the challenging data sets like NIPS 2003 datasets and Microarray dataset. The data was

meant to instantiate domains involving thousands of features. These domains contain a substantial proportion of irrelevant features that provides adequate scope to test the proposed feature selection method. The findings of the experiments carried out were consistent within the expected efficacy of the feature selection method proposed showing that the feature selection improved the accuracy.

The objective of this thesis was to develop a self-organized hierarchical learning neural network. From the experimental evaluation of the PURE feature selection methods above, the PURE feature selection method was used as a pre-processing procedure for the HieFLANN and HieFLANN_BP algorithm. The PURE feature selection method was able to facilitate the HieFLANN and HieFLANN_BP by providing only useful feature subset.

6.6 INVESTIGATION OF HIEFLANN ON HIGH DIMENSIONALITY DATASETS

The following experiments were carried out to investigate the feasibility of the HieFLANN in domains with large feature dimensionality.

Dataset	d	SF	Original Feature Space HieFLANN (%)	Reduced Feature Space HieFLANN (%)
Segment	19	7	82.26	79.56
Ionosphere	34	16	83.40	85.07
Dermatology	34	11	85.40	83.82
Sonar	208	11	75.19	77.50
Multi feature	649	126	10.01	74.08

Table 6-10 Accuracy of HieFLANN for UCI datasets in the original feature space and reduced feature space

Table 6-10 shows the accuracy for UCI datasets with and without PURE feature selection. For the datasets used, the accuracy obtained by the HieFLANN in the reduced feature space is comparable with the original feature space. It is noticed that the HieFLANN is not able to function correctly with Multi Feature dataset. The was probably due to the

failure of the eigen analysis process when the CCM undergoes matrix inversion. Dataset containing highly irrelevant features are likely to cause the CCM matrix to become a sparse matrix. The irrelevant feature yielded approximately zero correlation coefficients. The sparse CCM causes round off errors that led to incorrect eigenvectors derivations. As a consequent the C2FeCom compression that relies on the eigenvector analysis was likely to have failed and this led to poor results. The HieFLANN classification process is also likely to be hindered when SF/d is small in value. For this reason the testing for the HieFLANN using Microarray and NIPS 2003 datasets were not performed in their original feature space.

Dataset	d	SF	SF/d
Colon Tumor	2000	32	0.01600
Leukemia (aml, all)	7129	1	0.00014
Lung cancer	12533	16	0.00128
Prostate cancer	12600	1	0.00008
Ovarian cancer	15154	6	0.00039

Table 6-11 SF/d ratio for Microarray datasets

Table 6-11 record the SF/d ratio for Microarray datasets, the SF is according to the PURE method. It is noticed that all the SF/d values were small values close to zero. The observation indicates that it is better to have feature filtering pre-processing prior to the classification process.

A further investigation was to examine the performance of the HieFLANN with the PURE feature selection as pre-processing step. This yielded results in Table 6-12. The first column shows the datasets. The second column indicates the HieFLANN's accuracy on the reduced feature space. The third column is the accuracy recorded using C4.5 algorithm on the reduced feature space and last column is the difference in the accuracy obtained using HieFLANN and C4.5.

Dataset	HieFLANN (%)	C4.5 (%)	HieFLANN-C4.5 (%)
Colon Tumor	84.45	80.00	4.45
Lung cancer	89.93	88.50	1.43
Ovarian cancer	93.33	95.55	-2.22

Table 6-12 Accuracy of the HieFLANN compared to C4.5 with PURE feature selection

The results showed that the HieFLANN with ‘filtered’ feature space performed comparably as the C4.5 classification model. As the Leukemia (aml, all) and Prostate cancer datasets had only a single feature selected, it was not possible to implement using the HieFLANN. The PURE feature selection process filtered out irrelevant features and it provided a cleaned set of data for the HieFLANN. As expected, the HieFLANN did not work properly when the SF/d value was small. This is seen in the Colon Tumor dataset where 32 features were selected out of total 2000.

6.7 INVESTIGATION OF HIEFLANN_BP ON HIGH DIMENSIONALITY DATASETS

In this section, experiments were conducted with the presentation of domains having high proportion of irrelevant features.

Dataset	C4.5 (%)	HieFLANN_BP (%)	HieFLANN_BP - C4.5 (%)
Segment	89.25	80.66	-8.59
Ionosphere	85.32	84.10	-1.22
Dermatology	93.99	93.99	0.00
Sonar	70.92	80.00	9.08
Multi Feature	93.13	76.66	-16.47

Table 6-13 Accuracy of HieFLANN_BP in the original feature space compared to C4.5 in the reduced feature space using PURE for UCI datasets

The experimental results obtained using the HieFLANN_BP in the original feature space compared to the accuracy obtained using C4.5 algorithm in the reduced feature space were shown in Table 6-13. C4.5 was implemented in the Weka platform. The accuracies were not as good as the one with feature selection prior the classification process. This is shown in the last column on the difference of the accuracy. This phenomenon was expected for the domains with large feature dimensionality where significance is

restricted to only a small subset of the features. The MLP type network with Backpropagation learning algorithm has a worst case sample complexity that grows at least linearly in the number of irrelevant features [6]. In the reference, the author commented that in supervised learning with many input features, over fit is usually a potential problem unless there are ample training samples.

The performance of the HieFLANN_BP was tested using the domains with PURE feature selection method. Vigilance setting of the KFLANN was set at 0.9 and the feature tolerance is set using the standard deviation. The MLP network (MISO structure) was built with a single hidden layer and number of hidden neuron was set according to standard practice as in Appendix B.

Dataset	d	SF	HieFLANN_BP (%)	
			Original Feature Space	Reduced Feature Space
Segment	19	7	80.66	79.28
Ionosphere	34	16	84.10	84.80
Dermatology	34	11	93.99	89.30
Sonar	208	11	80.00	69.62
Multi feature	649	126	76.66	92.98

Table 6-14 Accuracy obtained using HieFLANN_BP with PURE feature selection for UCI datasets

The results of the HieFLANN_BP's performance with reduced feature space for UCI datasets was tabulated in Table 6-14. The comparison of the HieFLANN_BP with and without the PURE as pre-processing procedure was shown in Figure 6.4, it is noticed that the accuracies are comparable. The advantage is then in the use of a smaller feature set for discrimination.

Figure 6.4 Plot of the accuracy in the original feature space and reduced feature space

The reduced feature space contain only features that were important for class discrimination, as compared to the original feature space that contaminate with the irrelevant features that may hinder the class analysis process.

Dataset	HieFLANN_BP (%)	C4.5 (%)	HieFLANN_BP-C4.5 (%)
Colon Tumor	71.43	80.00	-8.57
Lung cancer	90.70	88.50	2.20
Ovarian cancer	97.20	95.55	1.65

Table 6-15 Accuracy of HieFLANN_BP compared to C4.5 (both with PURE feature selection) for Microarray datasets

The results obtained using the HieFLANN_BP for Microarray datasets with the PURE feature selection as the pre-processing step, are shown in Table 6-15. The difference between the accuracy of the HieFLANN_BP and the C4.5 is indicated in the last column. C4.5 performed better than HieFLANN_BP for Colon Tumor dataset. For Lung Cancer and Ovarian Cancer, the HieFLANN_BP obtained higher accuracy compared to C4.5.

6.8 CHAPTER SUMMARY

In this chapter, the PURE feature selection method was evaluated using three categories of datasets. The experimental results indicate that PURE was a reliable method for feature selection. As a pre-processing step to the C4.5 classification algorithm PURE improved

the accuracy. When compared with other methods (such as Correlation Feature Selection, Consistency and ReliefF), PURE exhibited comparable accuracy. The objective of this thesis is to have a self-organized hierarchical learning neural network. The purpose of PURE was to use it as a pre-processing step that facilitates the hierarchical learning neural network in its learning process.

Domains that have high proportional of the irrelevant features within the entire feature space tend to hinder learning algorithms. It is often necessary to filter the feature space to remove irrelevant features to provide a more efficient learning process. When the HieFLANN cluster the domains with high proportional of irrelevant features, as seen in Section 6.6, it cannot work properly. When the PURE feature selection method was applied to those domains, the HieFLANN worked efficiently and the experimental evidence indicated good results. A similar situation for the HieFLANN_BP, the PURE feature selection method assisted the learning process.

Relevance removes the irrelevant features and eliminates redundant features. Feature purity extracts feature that are the most significant in representing classes. A complementary coverage of the proposed hybrid feature evaluation index relevance and purity enjoy the merits of both indexes while reducing the shortcomings. The proposed filtered based feature selection scheme exhibits desirable properties. This includes the following:

- It does not rely on any searching strategy to select optimal feature subsets.
- It is simple to implement.
- It is fast to obtain the result.
- It is not affected by any bias of a learning algorithm.

The feasibility of the proposed feature evaluation index was established experimentally. It was also demonstrated through extensive experiments that the PURE feature evaluation was able to remove useless features while preserving the discriminative information faithfully. The proposed feature evaluation index accounts for both numerical and nominal type features and accommodates for most problem domains in the real world. When compared with other feature selection methods, the proposed scheme showed superior evaluation accuracy.

In section 6.4.2, comparative study between PURE feature selection method and other well known feature selection methods (entropy based, chi squared and correlation based) have been performed. Characteristics of the compared feature selection methods show that PURE feature selection method is less complex as compared to all compared methods. When Chi Squared feature selection is used, we need to discretize numerical (continuous) features that increase the complexity of the feature selection process. Correlation based feature selection lies heavily on the searching paradigm employed, parameter such as starting point, searching strategy, termination criteria determine the successful of the searching scheme. It is unlikely to use correlation based method for large dimensional problem set due to the exhaustive search for possible feature subsets is prohibitive. However this does not applicable to the PURE feature selection method.

7 CONCLUSION AND DISCUSSION

This chapter summarizes the research work needed to develop Hierarchical Neural Network structures. The emphasis of this thesis was on developing a system that would address issues on the curse of dimensionality using a hierarchical approach and dedicated processing pathways such as those evident within the human brain. It was necessary to develop modular networks that could effectively represent biologically equivalent neural pathways. This thesis presented 4 main chapters that contribute to the knowledge of this cause. Respectively from Chapters 3 to 6, these contributions were in:

- (a) Improving the Fast Learning Artificial Neural Network clustering algorithm to overcome its limitation of cluster inconsistent and sensitivity to data pattern sequence.
- (b) Developing unsupervised hierarchical clustering network Hierarchical Fast Learning Artificial Neural Network and using CCA to build a compression technique known as C2FeCom.
- (c) Integrating KFLANN with the MLP-BP type network to create a hybrid system. This was used as an atomic module to develop the self-supervised, learning hierarchical network – HieFLANN_BP.
- (d) Introducing the Purity and RElevancy (PURE) feature selection method to facilitate the filtering and cleaning of raw data.

7.1 CONTRIBUTIONS

From the experimental investigation performed in Chapter 3, the tolerance setting using the standard deviation was shown to be able to obtain an optimal performance for the KFLANN algorithm. While this was not the only way of setting the tolerance value, it provided a stable platform for autonomous determination of network tolerance parameters.

The cluster consistency of the KFLANN algorithm was also investigated through experimental testing and the results showed that the proposed data pattern reshuffling method was able to provide consistent performance that resolved cluster inconsistencies. The KFLANN was able to stabilize within a small number of iterations, allowing the consistent clusters to emerge within 5 iterations.

The end of Chapter 3 presented a discussion and experimental results analysis of the Mahalanobis distance method used in the KFLANN. It showed the ability to boost the accuracy when the domain contained correlated features. When the domain does not have any correlated features, both the Mahalanobis distance method and Euclidean distance methods become equivalent in the accuracy.

The work in Chapter 3 was crucial in developing a reliable platform that could cluster consistently, making it dependable for stereotyping of class topologies. It was then used to build other hierarchical structures as it was capable of autonomous configurations based on the dataset characteristics. This autonomous configuration was essential because in hierarchical layers, it is sometimes not possible to enter into intermediate levels to manually configure the networks. Thus, an autonomous configuration was a suitable means to meet the objectives.

Chapter 4 introduced the hierarchical based clustering model known as Hierarchical Fast Learning Artificial Neural Network. The HieFLANN actualizes the divide-and-conquer concept and provides a means to overcome the curse of dimensionality. This hierarchical method was used to overcome the limitation of the non-hierarchical (flat) clustering algorithm when handling large dimensional domains. This scheme was developed as a result of inspirations gained from the way biological neural system process information.

The concept was essentially to partition the large problem into smaller sub-problems and solve individual parts using small localized models. The final solution was the fusion of the sub- solution spaces to resolve the complete problem. The HieFLANN was presented as a unified framework that performs feature transformation and clustering in its autonomously derived hierarchical structure. It presents an efficient hierarchical network construction method based on the homogenous feature subspace development. The feature reduction concept in the HieFLANN was based on the Canonical Correlation Analysis method and was referenced as Canonical Covariance Feature Compression. The C2FeCom transformed the original feature space onto the compressed form feature space which aimed to maximize the preservation of the original information.

The results obtained from the experimental evaluation on the HieFLANN using real life benchmark datasets showed that the totally unsupervised HieFLANN network was able to cluster the given datasets. When compared to non-hierarchical KFLANN networks, it also showed that the HieFLANN produced better accuracy for most of the datasets. The scalability of the HieFLANN was also compared to its non-hierarchical KFLANN. The results showed that HieFLANN was able to scale up its processing with the increase in number of data patterns and number of dimensionality. In general, as compared to KFLANN, HieFLANN performed better.

Clustering plays an important role to extract hidden features as well as for the interpretation of discovered information. However clustering algorithms are not real learning systems unless they can adapt. Chapter 5 represents the push towards the goal of emulating the way human brain processes information. It was necessity to develop a more robust and efficient learning system, rather than a rigid clustering algorithm. This led to the development of hybrid learning model that consisted of the KFLANN network as the

first stage clustering system and the MLP-BP being, the classifier and universal function approximator for the second stage. The hybridization of these two networks saw the creation of new paradigm. Experimental studies using standard benchmark datasets indicated that the Hierarchical structure model with the hybrid KFLANN-BP was efficient and accurate. While the HieFLANN in Chapter 4 would become cumbersome when the feature dimensionality grows beyond tens of thousands due to the heavy computation involved in the C2FeCom, this was not present in the hybrid learning model known as HieFLANN_BP. The structure mimics the HieFLANN except each subnet is replaced with a hybrid learning model (KFLANN_BP) and there was no need for the CCA feature compression processes.

Apart from the benefits discussed, the theoretical difference between the single calibrated KFLANN and the hybrid KFLANN_BP were discussed. The experimental results obtained favored the self-supervised hybrid learning KFLANN_BP as it performed better than its single calibrated unsupervised KFLANN.

Chapter 6 dwelled with the need for relevant feature spaces. Despite the availability of efficient algorithms, dealing with arbitrary large dimensional problem domains still requires the screening of irrelevant features. The algorithm still suffers from poor performances due to the presence of the noisy or irrelevant features. This led to the suggestion of a feature selection scheme based on the feature PUrity and RElevancy (PURE) to facilitate the network learning task. The proposed method PURE feature scheme ranks based on feature purity in distinguished classes and the relevance (inter-feature and feature-classes). It subsequently selects a subset of most important features from the ranked list.

The experimental results indicated that the proposed feature selection method PURE was a reliable method for feature selection. An overall comparison was done between the PURE with other methods such as Correlation Feature Selection, Consistency and ReliefF. The results indicated that PURE exhibited comparable accuracy.

Since problem domains with a high proportion of irrelevant features tend to hinder the proposed learning algorithms, it was necessary to filter the feature space to remove useless features. The investigation showed that the HieFLANN did not perform well in domains with many irrelevant features. A similar encounter was met for the HieFLANN_BP. However the PURE feature selection method assisted both the learning processes of both methods positively.

The results from the full construction of the HieFLANN-BP indicate a possibility of self-supervision, using KFLANN to derive clusters from the data. Parallelism is possible in HieFLANN-BP but not for the non-hierarchical MLP-BP (or KFLANN). Hybridization of the KFLANN and MLP-BP is advantageous. HieFLANN-BP inherits the MLP-BP's capability to handle overlapping class distribution by skewing its manifold topology. MLP-BP neural model formularizes the optimization function of a given problem set, this beneficial to the HieFLANN-BP. Cluster stabilization from the KFLANN guarantees accurate generalization for the MLP-BP network, indirectly attain to stability of HieFLANN-BP.

The use of the hybridization process just illustrates the possibility of combining KFLANN with another learning paradigm. It is indeed possible to incorporate other learning architectures. Through the use of the autonomous clustering KFLANN paradigm, it is possible to partition problem spaces into manageable smaller problem subspace. The

road to viable long term learning models needs to utilize this concept of divide and conquer to build suitable memory models for future long term learning cognitive systems. The presentation of the work in this thesis, not as a means to neither compare network accuracies nor challenge existing network structures, but as an attempt to present biologically inspired principles that can extend current networks into the hierarchical domain.

7.2 FUTURE DIRECTIONS

The KFLANN underwent minor but significant modifications and is able to cluster more effectively. The new data pattern reshuffling and Mahalanobis distance metric improved the KFLANN's performance. To date the feature tolerance setting determines the final clustering results based only on the standard deviation method. Although this approach works well for most of the datasets, sparse datasets may hinder its functions. Further investigation can be made to determine other heuristics for tolerance and vigilance setting.

Some issues still remain unresolved in the proposed PURE feature selection methodology. Since this was not the focus of the thesis, it is left to be engaged by other future researchers. The time comparisons cannot be made as there are no known available references that recorded the necessary timing details.

The proposed feature selection scheme can be further investigated for parallel implementations. Scalable feature selection may be extended to distributed or parallel computing that favors the real life applications. Since the proposed feature evaluation index is computed on each feature independently except for inter-feature relevancy, it has good potential to be implemented in parallel.

From this thesis, 5 conference papers were published. Three were published in IJCNN and two in ICCNS held in Boston University. Another two manuscripts were submitted to IEEE Transactions on Neural Networks and are still undergoing review.

REFERENCE

- [1] Aapo Hyvarinen, Karhunen J. and Oja E. (2001). Independent component analysis. John Wiley & Sons, New York.
- [2] Ackley D. H., Hinton G. E. and Sejnowski T. J., (1985). A learning algorithm for Boltzmann machines. *Cognitive Science* 9:147-169.
- [3] Aha D. W. and Bankert R. L., (1995). A comparative evaluation of sequential feature selection algorithms. *Proceeding of the fifth International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL.
- [4] Alex Tay L. P. and Prakash S., (2002). K-Means Fast Learning Artificial Neural Network, an alternate network for classification. *Proceedings of 9th International Conference on Neural Information Processing* 2:925-929.
- [5] Anderson J. A., (1972). A simple neural network generating an interactive memory. *Mathematical Bioscience* 14:197-220.
- [6] Andrew Y. N., (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada.
- [7] Arfken G., Lagrange multiplier. *Mathematical Methods For Physicists* (3rd edition). Academic Press, Orlando FL. 945-950.
- [8] Ariew R., (1976). Ockham's Razor: A historical and philosophical analysis of Ockham's principle of parsimony. *Philosophy*. Champaign-Urbana, University of Illinois.
- [9] Ben-Bassat M., (1982). Use of distance measures, information measures and error bounds in feature evaluation. *Classification, Pattern Recognition and Reduction of Dimensionality: Handbook of Statistics*, P. R. Krishnaiah and

L. N. Kanal, Editors, The Netherlands: North-Holland, Amsterdam, 2: 773–791.

- [10] Blaz Novak (2004). Use of unlabeled data in supervised machine learning. *Conference on Data Mining and Warehouse (SiKDD 2004)*, Slovenia.
- [11] Blum A. and Langley P., (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97: 245-271.
- [12] Borga M., (2001). Canonical correlation a tutorial. [<http://people.imt.liu.se/~magnus/cca/>]
- [13] Cao L. J., Keerthi S. S., Ong C. J., Uvaraj P., Fu X. J., Lee H. P., (2006). Developing parallel sequential minimal optimization for fast training support vector machine, *Neurocomputing* 70, 93-104.
- [14] Carpenter G. A. and Grossberg S., (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, 37:54-115.
- [15] Carpenter G. A. and Grossberg S. and Gregory W. Leshert, (1998). The what and where filter, a spatial mapping neural network for object recognition and image understanding. *Computer Vision and Image Understanding* 69(1):1-22.
- [16] Carpenter G. A. and Grossberg S., (2001). Adaptive resonance theory. The handbook of brain theory and neural networks, second edition, MIT Press.
- [17] Cervera E. and Pobil A. P. del, (1997). Multiple self-organizing maps: A hybrid learning scheme. *Neurocomputing*, 16(4):309-318.
- [18] Cho Siu-yeung, Chow T. W. S. (1999). Training multiplayer neural networks using fast global learning algorithm – least squares and penalized optimization methods. *Neurocomputing* 25, 115-131.

- [19] Churchland P. S. and Sejnowski T. J., (1992). The computational brain. MIT Press, Cambridge, MA.
- [20] Cover T. M. and Van Campenhout J. M., (1977). On the possible ordering in the measurement selection problem. *IEEE Transaction Systems, Man And Cybernetic* 7, 657-661.
- [21] Christopher M. Bishop, (1995). Neural networks for pattern recognition. Oxford University Press, 6-7.
- [22] Dash M. and Liu H., (1997). Feature selection for classification. *Intelligent Data Analysis* 1(3).
- [23] Deng D., Kasabov N., (2000). ESOM: an algorithm to evolve self-organizing maps from on-line data streams. *Proceedings of IJCNN 2000* 4, 3-8.
- [24] Deng D., Kasabov N. K., (2003). On line pattern analysis by evolving self-organizing maps. *Neurocomputing* 51, 87-103.
- [25] Dittenbach M., Merkl D. and Rauber A. (2000). The Growing Hierarchical SOM (GHSOM). *International Joint Conference On Neural Network*, Como, Italy.
- [26] Doak J., (1992). An evaluation of feature selection methods and their application to computer security. *Technical Report CSE 92-18*, Department of Computer Science and Engineering, University of California, Davis.
- [27] Dong J. X., Adam K. Ching Y. S. (2003). A fast parallel optimization for training support vector machine, Lecture note in computer science, P. Perner and A. Rosenfeld Eds: MLDM 2003, LNAI 2734, 96-105.
- [28] Duda R. O. and P. E. Hart, (1973). Pattern classification and scene analysis. New York: John Wiley & Sons.

- [29] Eric de Bodt, Marie Cottrell and Michel Verleysen, (1999). Using Kohonen algorithm for quick initialization of simple competitive learning algorithm. *Proceeding of the European Symposium on Artificial Neural Networks ESANN'99*, 19-26.
- [30] Evans D. J., Tay L. P., (1995). Fast Learning Artificial Neural Network for continuous input applications. *Kybernetes* 24(3).
- [31] Edwards A. L., (1976). The correlation coefficient. Chapter 4 in *An Introduction to Linear Regression and Correlation*, San Francisco, CA: W. H. Freeman, 33-46.
- [32] Fausett L., (1994). Fundamentals of neural network: architectures, algorithms and applications. Englewood Cliffs, NJ: Prentice Hall.
- [33] Fayyad U. and Irani K. (1993). Multi interval discretization of continuous valued attributes for classification learning, *Proceeding 13th International Joint Conference on Artificial Intelligence*.
- [34] Frasconi P., Gori M. and Soda G., (1997). Links between LVQ and Backpropagation. *Pattern Recognition Letters*, 18(4):303-310.
- [35] Fu X. J., Wang L. P., (2003). Data dimensionality reduction with application to simplifyinh RBF network structure and improving classification performance, *IEEE Transactions on System, Man, Cybernatic Part B – Cybernatic* 33, 399-409.
- [36] Fukushima K., Miyake S. (1982). Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition* 15(6), 455-469.
- [37] Fukushima K., Miyake S. and Ito T., (1983). Neocognition: a neural network model for a mechanism of visual pattern recognition. *IEEE Transaction On Systems, Man And Cybernetics*, SMC-13 (5):826-834.

- [38] Gersho A. and Gray R. M., (1991). Vector quantization and signal compression. Kluwer Academic, Boston, MA.
- [39] Grossberg S., (2003). How does the cerebral cortex work, development, learning, attention and 3D vision by laminar circuits of visual cortex. *Technical Report CAS/ CNS TR-2003-005*.
- [40] Grossberg S., (1976). Adaptive pattern classification and universal recording, I: parallel development and coding of neural feature detector & II: feedback, expectation, olfaction and illusion. *Biological Cybernetics*, 23:121-134 & 187-202.
- [41] Grossberg S., (1970). Neural pattern discrimination. *Journal of Theoretical Biology*, 27:291-337.
- [42] Hall M. A. and Smith L. A., (1999). Correlation based feature selection for machine learning: comparing a correlation based filter approach to wrapper. *Proceeding of the Florida Artificial Intelligence Symposium*.
- [43] Hartigan J., (1975). Clustering algorithms. New York: John Wiley & Sons.
- [44] Hebb D. O., (1949). The organization of behaviour. New York: Wiley.
- [45] Hecht-Nielsen R., (1987). Kolmogorovo's Mapping neural networks existence theorem. *Proceedings of the First IEEE International Conference on Neural Networks*, 3:112-114.
- [46] Hecht-Nielsen R., (1989). Theory of Backpropagation neural networks. *Proceedings of the International Joint Conference on Neural Networks*, 1:593-605.
- [47] Henry A. Ruger and Clara E. Bussenius (1913). Memory: A contribution to experimental psychology by Hermaan Ebbinghaus (1885). Teachers College, Columbia University, New York.

- [48] Hicks J. and Monaghan P., (2001). Explorations of the interaction between split processing and stimulus types. In Wermter S., Austin J., Willshaw D., editors. *Emergent Neural Computational Architectures based on Neuroscience*, Springer-Verlag, Heidelberg, Germany.
- [49] Hodgkin A. L. and Huxley A. F., (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology* 117:500-544.
- [50] Hopfield J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy of Science of the USA* 79: 2554-2558.
- [51] Huber P. J. (1985). Projection pursuit. *The Annals Of Statistical*, 13 (2):435-475.
- [52] Ian H. Witten and Eibe Frank (2005). Data mining: practical machine learning tools and techniques. 2nd edition Morgan Kaufmann, San Francisco.
- [53] Jain A. K. and Dubes R. C. (1988). Algorithms for clustering data. Prentice Hall advanced reference series, Upper Saddle River, New Jersey.
- [54] Jain A. K., Murty M. N. and Flynn P. J. (1999). Data clustering: a review. *Association for Computer Machinery Computing Surveys*, 31(3):264-323.
- [55] Jain A. K., Robert P. W. Duin and Mao J. C. (2000). Statistical pattern recognition: a review. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 22 (1):4-37.
- [56] James A. Reggia, Yuri Shkuro and Natalia Shevtsova (2001). Computational investigation of hemispheric specialization and interactions. In Wermter S., Austin J., Willshaw D., editors, *Emergent Neural Computational Architectures based on Neuroscience*, Springer-Verlag, Heidelberg, Germany.

- [57] John von Neumann (1958). The computer and the brain. New Haven, London, Yale University Press.
- [58] Jolliffe I. T. (2002). Principal component analysis. Springer series in Statistical, Springer-Verlag, New York, 2nd Edition.
- [59] Kasabov N., Song Q. (2001). DENFIS: Dynamic evolving neural-fuzzt inference system and its application for time series prediction. *IEEE Transactions on Fuzzy Systems*.
- [60] Kaski S. and Kohonen T. (1994). Winner Take All networks for physiological models of competitive learning. *Neural Networks* 7:973-984.
- [61] Kaufman L. and Rousseeuw P. J. (1990). Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, New York.
- [62] Kimura S., Ozawa S., Abe S., (2005). Incremental kernel PCA for online learning of feature space. Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, control and automation and international Conference On Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05).
- [63] Knoblauch A. and Palm G. (2001). Spiking associative memory and scene segmentation by synchronization of cortical activity. In Wermter S., Austin J., Willshaw D. (eds). *Emergent Neural Computational Architectures based on Neuroscience, LNAI 2036 Springer-Verlag Heidelberg*, 407-427.
- [64] Kohonen T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59-69.
- [65] Kohonen T. (1986). Learning vector quantization for pattern recognition. Technical Report TKK-F-A601, Helsinki University of Technology, Espoo, Finland.

- [66] Kohonen T. (1988). Learning vector quantization. *Neural networks*, 1 (suppl.1), 303.
- [67] Kohonen T. (1988). The self-organizing and associative memory. Springer Verlag.
- [68] Kohonen T. (1990). The self-organizing map. *Proceedings of the IEEE* 78, 1464-1480.
- [69] Kohonen T. (2001). Self-organizing Maps, third edition, Springer.
- [70] Kononenko I. (1995). On biases in estimating multi-valued attributes. *Proceeding Of International Joint Conference on Artificial Intelligence*, 1034–1040.
- [71] Kozma R. and Freeman W. J. (2003). Basic principles of the KIV model and its application to the navigation problem. *Journal of Integrative Neuroscience* 2 (1):125-145.
- [72] Kurt Hornik, Maxwell Stinchcombe and Halbert White (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359-366
- [73] Lashley K. (1950). In search of the engram. *Symposia Society of Experimental Biology*, 4: 454-482.
- [74] Lee D. S., Srihari S. N. and Gaborski R. (1991). Bayesian and neural network pattern recognition: Theoretical connection and empirical results with handwritten characters. *I. Sethi and A. Jain, Eds. Artificial Neural Networks and Statistical Pattern Recognition*.
- [75] Liu H. and Setinno R. (2001). Chi2: Feature selection and discretization of numeric attributes, *Proceeding IEEE 7th International Conference on Tools with Artificial Intelligen*, 338-391.

- [76] Liu S., Yi H., Chia L. T., Rajan D., (2005). Adaptive hierarchical multi-class SVM classifier for texture-based image classification. *IEEE International Conference on Multimedia and Expo*.
- [77] Louie K. and Wilson M. A., (2001). Temporally structured REM sleep replay of awake hippocampal ensemble activity. *Neuron*, 29:145-156.
- [78] Lovell D. R., Downs T. Tsoi A. C., (1997). An evaluation of the neocognitron. *IEEE Transactions on neural networks* 8 (5).
- [79] MacQueen J. B., (1967). Some ,Methods for classification and analysis of multivariate observations. *Proceedings of fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1:281-297.
- [80] Mahalanobis P. C., (1936). On the generalized distance in statistic. *Proceedings of Natural Institute of Science* 2:49-55.
- [81] Martinex A. M., Kak A. C., (2001). PCA versus LDA. *IEEE Transactions on Pattern analysis and machine intelligence*, 23 (2), 228-233.
- [82] Masshi-Reza Amini and Patrick Gallinari, (2002). The use of unlabeled data to improve supervised learning for text summarization. *Annual ACM Conference on Research and Development in Information Retrieval, Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 105-112.
- [83] Macquet P., Laureys S., Peigneux P., Fuchs S., Petiau C., Philips C. and Aerts J., (2000). Experience-dependent changes in cerebral activation during human REM sleep. *National Neuroscience* 3,831-836.
- [84] McCulloch W. S. and Pitts W., (1943). A logical calculus and the ideas immanent in the nervous activity. *Bulletin of Mathematical Biophysics* 5(115-133):18-27.

- [85] Merz C. J. and Murphy P. M., (1996). UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mlearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science.
- [86] Meyer D. E., Mueller S. T., Seymour T. L. and Kieras D. E., (2000). Brain loci of temporal encoding and serial order control for verbal working memory revealed by computational modeling and focal lesion analysis of memory span performance. *Cognitive Neuroscience Society Annual Meeting Program*, 147.
- [87] Minsky M. and Seymour Papert, (1969). Perceptrons: An introduction to computational geometry. MIT Press, Cambridge.
- [88] Mishkin M., Unterleider L. G. and Macko K. A., (1982). Object vision and spatial vision: two cortical pathways. *Trends in Neuroscience* 6, 414-417.
- [89] Monoranjana D., Liu H. and Motoda Hiroshi, (2000). Consistency based feature selection source. *Lecture notes in Computer Science, Proceeding of the 4th Pacific-Asia Conference On Knowledge Discovery And Data Mining* 1805:98-109.
- [90] Monoranjana D. and Liu H., (2003). Consistency based search in feature selection. *Artificial Intelligence*, 151(1-2):155-176.
- [91] Narendra P. M. and Fukunaga K., (1977). A branch and bound algorithm for feature subset selection. *IEEE Transaction on Computers*, C-26 (9), 917-922.
- [92] Nigam K., Andrew McCallum, Sebastian Thrun and Tom Mitchell, (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39(2/3), 103-134.
- [93] Nikola Kasabov, (2001). Evolving fuzzy neural networks for supervised/unsupervised online knowledge based learning. *IEEE Transactions of Systems, Man And Cybernetics, Part (B)* 31(6):902-918.

- [94] Oh I. S., Lee J. S., Moon B. R., (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions of Pattern Analysis and Machine Intelligence* 26, 1424-1437.
- [95] Ozawa S., Pang S., Kasabov N., (2006). An incremental principal component analysis for chunk data. *IEEE International Conference on Fuzzy System*.
- [96] Ozawa S., Pang S., Kasabov N., (2006). On-line feature selection for adaptive evolving connectionist systems. *International Journal of Innovative Computing, information and control* 2 (1).
- [97] Pang S., Ozawa S. and Kasabov N., (2005). Incremental Linear Discriminant Analysis for Classification of data streams. *IEEE Transactions on System, Man and Cybernetics-Part B: Cybernetics*, 33 (5).
- [98] Papoulis A., (1984). Probability, random variables and stochastic processes. 2nd Edition, New York, McGraw Hill, 149-151.
- [99] Platt J., (1991). A resource allocating network for function interpolation. *Neural Computational* 3(2), 231-225.
- [100] Quinlan J. R., (1993). C4.5: Programs for machine learning. San Mateo, California, Morgan Kaufmann.
- [101] Rao C. R., (1973). Linear statistical inference and its applications. John Wiley & Sons, 582-585.
- [102] Raymer M. L., Punch W. F., Goodman E. D., Kuhn L. A., Jain A. K., (2000). Dimensionality reduction using genetic algorithms, *IEEE Transactions Evolutionary Computation* 4, 164-171.

- [103] Riga T., Cangelosi A. and Greco A., (2004). Symbol grounding transfer with hybrid self-organizing/supervised neural networks. *International Joint Conference on Neural Networks*.
- [104] Ripley B. D., (1996). Pattern recognition and neural network. Cambridge: Cambridge University Press.
- [105] Rissanen J., (1978). Modeling by shortest data description. *Automatica*, 14:165-478.
- [106] Ronan G. Reilly, (2001). Collaborative cell assemblies: building blocks of cortical computation. *Emergent Neural Computational Architectures based on Neuroscience*. Springer-Verlag.161-173.
- [107] Robnik-Sikonja M. and Kononenko I., (2003). Theoretical and empirical analysis of ReliefF and ReliefF. *Machine Learning*, 53(1-2):23-69.
- [108] Rosenblatt F. (1957). The perceptron: a perceiving and recognizing automation (project PARA). *Technical report* 85-460-1, Cornell Aeronautical Laboratory, Ithaca, New York.
- [109] Rosenblatt F., (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386-408.
- [110] Ruck D. W., Roger S. K., Kabrisky M., Oxley M. E., Suter B. W., (1990). The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks* 1(4):296-298.
- [111] Rui Xu, Wunsch D., (2005). Survey of clustering algorithm, *IEEE Transactions on Neural Network* 16 (3), 645-678.
- [112] Rumelhart D. E. and McClelland J. L., (1986). Parallel distributed processing, explorations in the microstructure of cognition. Cambridge, MA: MIT Press.

- [113] Scholkopf, Smola A. and Muller K. R., (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10 (5), 1299-1319.
- [114] Sun R., (1996). Hybrid connectionist symbolic models. *Artificial Intelligence Magazine* 17(2), 99-103.
- [115] Taguchi G. and Jugulum R., (2002). The Mahalanobis-Taguchi strategy, a pattern technology system. John Wiley & Sons, New York.
- [116] Tay L. P. and Evans D. J., (1994). Fast Learning Artificial Neural Network (FLANN II) Using Nearest Neighbors Recall. *Neural Parallel and Scientific Computations*, 2 (1).
- [117] Tsao E. C-K., Bezdek J. C. and Pal N. R., (1994). Fuzzy Kohonen clustering networks. *Pattern Recognition*, 27 (5):757-764.
- [118] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth and Ramasamy Uthurusamy G., (1996). Advances in knowledge discovery and data mining, AAAI/MIT Press, Menlo Park, CA.
- [119] Wang X., Paliwal Kuldeep K., (2003). Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition. *Pattern recognition* 36.
- [120] Welling M., (2006). Fisher linear discriminant analysis. http://www.ics.uci.edu/~welling/classnotes/papers_class/Fisher-LDA.pdf
- [121] Werbos P. J., (1974). The roots of backpropagation, NY: John Wiley & Sons.
- [122] White A. P. and Liu W. Z. (1994). Technical Note: Bias in information-based measures in decision tree induction. *Machine Learning*, 15:321–329.

- [123] Widrow B. and Hoff M. E., (1960). Adaptive switching circuits. *IRE WEBSCON Convention record*, 96-104.
- [124] William J., (1909). Pluralism, pragmatism and instrumental truth. *A Pluralistic Universe*, New York, 321-324.
- [125] Wilson M. A. and McNaughton B. L., (1994). Reactivation of hippocampal ensemble memories during sleep. *Science* 265 (5172), 676-679.
- [126] Wong L. P., Alex T. L. P. and Xu J., (2003). Centroid stability with K-Mean Fast Learning Artificial Neural Networks. *International Joint Conference On Neural Network*.
- [127] Wood S. and Gedeon T. D., (2001). A hybrid neural network for automated classification. *Proceeding of 6th Australian Document Computing Symposium*, Coffs Harbour, Australia.
- [128] Yang J., Yang X., Zhang J., (2006). A parellel multi class classification support vector machine based on sequential minimal optimization, *Proceedings of the first Internatioanal multi symposiums on computer and computatioanal sciences*.
- [129] Yule G. U., (1912). On the methods of measuring the association between two attributes. *Journal of the Royal Statistical Society*, 75:579-652.
- [130] Verikas A., Gelzinis A., Malmqvist K. and Bacauskiene M., (2001). Using unlabelled data to train a multilayer perception. *Neural Processing Letters*, 14(3):179-201.
- [131] Zhang P., Peng J. and Norbert R., (2005). Discriminant analysis: a unified approach, *5th IEEE International Conference on Data Mining*.

APPENDIX A – KFLANN CONVERGENCE

Since the combined distribution of any feature is not likely to be normal, the non-normal distributed dataset is assumed and the Chebyshev's Rule applies. For simplicity in discussions, subsequence descriptions and references are only made to single features. Even if multiple features are combined to discriminate classes the principle extends singularly across individual features.

The σ_o is defined as the standard deviation for any feature space of the entire dataset. As discussed in the random distribution using the Chebyshev 's rule, the σ_o will provide a subdivision of approximately 6 parts at 97% or 8 parts at 99% data space coverage. It is thus expected that under most natural occurring cases, this spread will provide individual clusters with its own coverage of the entire data space. It is also possible that some clusters may share common ranges that can only be discriminated by other feature definitions of the particular class. For example, a dataset with single feature and 3 classes (C_1 , C_2 and C_3) is shown in Figure A.1, the distance between the two nearest points of the different classes is $\text{Dist}(C_i, C_j)$.

Figure A.1 Nearest Distance between Nearest Classes And Overall Standard Deviation For A Single Feature Dataset Of 3 Classes

4 different types of groupings are used for the purpose of discussion in this thesis, showing how the standard deviation is used as feature tolerance setting that can lead to

convergence of the KFLANN clustering algorithm. Figure A.2 to Figure A.5 are dot plots of these 4 possible categories of datasets.

Figure A.2 Dot plot for case (A), normally distributed with equal standard deviation for well separable classes, $\text{Dist}(C_1, C_2)$ is greater than the σ_o

Figure A.3 Dot plot for case (B) normally distributed with unequal standard deviation for well separable classes, $\text{Dist}(C_1, C_2)$ is greater than the σ_o

Figure A.4 Dot plot for case (C), normally distributed with equal standard deviation for well separable classes, $\text{Dist}(C_i, C_j)$ are smaller than the σ_o

Figure A.5 Dot plot for case (D), normally distributed with unequal standard deviation for well separable classes, $\text{Dist}(C_i, C_j)$ are smaller than the σ_o

There are 2 classes for case A and B, each bell shape curve represent a single class. 3 classes for both case C and case D. The above cases are applicable for the domains that possess equal distributions between classes or bearing skewed class distributions. Artificial datasets were used to simulate the above 4 cases. The behavior of the KFLANN algorithm can be studied using these 4 synthetic datasets. The standard deviation method was used to set the tolerance value. All artificial datasets consist of single features. This use of a single feature is a means to convey the concept and it does not erode or devoid the need for multidimensional analysis. This portion provides the effects of tolerance over a single feature dimension.

A.1 Case (A) Well Separable Classes With Equal Standard Deviation With Distance Between Nearest Classes Are Greater Than σ_o

Figure A.6 Dot plot of dataset for case (A)

The dot plot of case (A) is shown in Figure A.6. Two distinct classes with twin peaks characterize a single feature. The $\text{Dist}(C_1, C_2)$ is 9 and the standard deviation for each class is equal with value 1. C_1 has the mean value at 4 and C_2 has the mean value at 20. Let the initial two data points in the data list be located at P_1 and P_2 . The σ_o was calculated to be 8. (Information for that calculation is not provided here as this section was meant to explain the shifting of centroids pictorially.)

Figure A.7 KFLANN clustering output at first iteration for case (A)

For this artificial dataset, the $\text{Dist}(C_1, C_2)$ is greater than the σ_o imply that the class separation is far apart. At the first iteration, two classes were grouped by two clusters as shown in Figure A.7. The area covered by the two clusters (with tolerance value at 8) overlaps (represent by the shaded part of the figure) with centroids at P_1 and P_2 .

At the second iteration, the respective centroids move towards to centers of central tendency within the bell curves. As compared to the first iteration, the two clusters were shifted further apart. This is illustrated in Figure A.8. The result is a wider divide between two clusters and the overlapping no longer exists after second iteration. Centroid P_1 and P_2 at the first iteration have moved to location as showed in Figure A.8.

Figure A.8 KFLANN clustering output at second iteration for case (A)

The clustering process will proceed to the third iteration due to the centroid movements and the final output is similar as the second iteration. Iterations then stop because there are no further changes in the centroids. The KFLANN clustering ends at the third iteration and 2 clusters are created.

A.2 Case (B) Well Separable Classes With Unequal Standard Deviation With Distance Between Nearest Classes Are Greater Than σ_o

As shown in dot plot of Figure A.9, artificial data in case (B) contains of 2 classes with normally distributed data, C_1 has mean value at 3 and C_2 has a mean value of 20. C_1 has a standard deviation 1 while C_2 has a standard deviation of 2. Let P_1 and P_2 be the first two data points located in the initial data list. The σ_o is 8.6576 and $\text{Dist}(C_1, C_2)$ is 9.

Figure A.9 Dot plot of data distribution for case (B)

After the first iteration of the KFLANN clustering, 3 clusters are formed as in Figure A.10 with centroids located at P_1 , P_2 and P_3 respectively. The individual cluster coverage is shown in the figure with overlaps between clusters. The data points closest to the centroids in each of the 3 clusters are shifted to the top of the new data list that is prepared for the next iteration. Cluster₁ with centroid located slightly toward the left side of the cluster, is expected to shift to left at the second iteration. This is because the mean tends to the left. Cluster₂ with centroid located toward the right side of the cluster, thus it is expected to shift to right at the second iteration as it has a mean tendency to the right.

Figure A.10 KFLANN clustering output at first iteration for case (B)

As a consequence of centroids shifting, 2 clusters are created at the second iteration and Cluster₃ at the first iteration is subsumed by Cluster₂, this is shown in Figure A.11 with centroid located at P_1 and P_2 , there is small overlap between the final two clusters.

Figure A.11 KFLANN clustering output at second iteration for case (B)

At the second iteration, the centroids movement leads to the third iteration of the clustering. The re-computation of centroids positions after second iteration shifts the centroid of Cluster₁ towards left side while Cluster₂ towards right side, the pictorial view is given in Figure A.12 without cluster overlapping with P_1 and P_2 as centroid.

Figure A.12 KFLANN clustering output at third iteration for case (B)

A.3 Case (C) Well Separable Classes With Equal Standard Deviation With Distances Between Nearest Classes Are Smaller Than σ_0

Figure A.13 Dot plot of dataset for case (C)

The Figure A.13 is the dot plot of the artificial data in case (C). There are 3 normally distributed classes as shown in the figure, with 3 well separated bell curves. C_1 has a mean value at 3, C_2 has 10 as its mean and 17 for C_3 . All classes have the same standard deviation value that is 1 while the σ_0 is 5.837. The $\text{Dist}(C_1, C_2)$ is 1.9862 and $\text{Dist}(C_2, C_3)$ is 1.2384. Both distances are smaller than σ_0 . The data pattern sequence follows the partitioning area as indicated by 1, 2, 3 and 4. The numerical order indicates the order of sequence. Assume that the initial clustering iteration started with data point located at P_1 and P_2 . This is an illustration of a poor case because the initial cluster created will center its reference points at P_1 and P_2 causing a half portion of the data points to belong to C_1 and the other half belong to C_2 . The other cluster will have half the portion of data points belonging to C_3 and the other half belong to C_2 . Figure A.14 shows the cluster formations after the first iteration with two clusters created. With data pattern reshuffling property, data points closest to centroids will be moved to top of the data list. As the consequence, Cluster_1 at the first iteration will shift toward the left while Cluster_2 (from the first iteration) will shift to right at the same iteration.

Figure A.14 KFLANN clustering output at first iteration

As shown in Figure A.15, Cluster₂ that was created at second iteration with centroid near the mean of C₂ (P₂), Cluster₂ centered near the mean of C₃ (P₃).

Figure A.15 KFLANN clustering output at second iteration

This leaves the middle section open to register new centroid points. After the second iteration, 3 clusters are created with centroids at P₁, P₂ and P₃ respectively. The cluster centroid re-computation and data pattern reshuffling will lead to the shifting of the

centroids. This can be seen as the centroid of Cluster₁ is located nearer to the left of the cluster as shown in Figure A.16 compared to second iteration and centroid of Cluster₃ located lean more to the right of the cluster compared to second iteration.

Figure A.16 KFLANN clustering output at third iteration

At the third iteration as shown in Figure A.16 three clusters are created. The centroid of the Cluster₁ stabilizes at the feature space it covers. The remaining portions of the data points are assigned to Cluster₂ and Cluster₃.

A.4 Case (D) Well Separable Classes With Unequal Standard Deviation With Distances Between Nearest Classes Are Smaller Than σ_0

Figure A.17 is the dot plot for artificial data with 3 classes of unequal standard deviations. Each bell curve dot plot group represents an individual class, C₁ has a mean value of 4, C₂ has mean value at 14 and C₃ a mean at 30. The σ_0 is 11.11 and Dist(C₁,C₂) is 6.5915 while Dist(C₂,C₃) is 7.2883.

Figure A.17 Dot plot for dataset for case (D)

All classes have different standard deviation values. σ_{C_1} is 1, 0.5 and 2.0 for σ_{C_2} and σ_{C_3} respectively. Again the order of data pattern sequence are shown as regions 1, 2, 3 and 4 which mean that the data point within the area 1 is the first portion of the data patterns presented follow by those in area 2, 3 and 4. P_1 and P_2 are the locations for the first two data points in the initial data list. For this dataset, at the first iteration 2 clusters are created as shown in Figure A.18.

Figure A.18 KFLANN clustering output at first iteration for case (D)

At the second iteration, the 2 clusters created are further apart as compared to first iteration shown in Figure A.19.

Figure A.19 KFLANN clustering output at second iteration for case (D)

It is noticed that these 2 clusters are stabilized at their current centroids. When two clusters are used to represent 3 classes, imply that the clusters are overlapped. One of the alternatives to handle such a case is to reduce the tolerance setting so that the cluster size is reduced with the hope that it can generate new set of cluster represent the existing classes.

For this case, the tolerance value is further multiply with 0.5. The original value of 11.11 , a multiplication will result in 5.55. Using P_1 and P_2 as start points as in Figure A.20, nearest data point to centroid of Cluster₁ (8.55) is located at P_1 while nearest data point to centroid of Cluster₂ is located at P_2 .

Figure A.20 Data pattern sequence for case (D) after second iteration

The KFLANN clustering process may start with the smaller tolerance at the third iteration or it can be executed from the beginning again. The clustering output of the third iteration is shown in Figure A.21. Cluster₁ and Cluster₂ are shifted to the left and right during the fourth iteration while Cluster₃ is created and it stabilizes immediately. While it is possible to begin immediately with 0.5, results may form more clusters.

Figure A.21 KFLANN clustering output at third iteration for case (D)

At the fourth iteration, 3 clusters are generated to represent the 3 classes as shown in Figure A.22. There were no data points existing at the overlapping area between Cluster₁ and Cluster₃.

Figure A.22 KFLANN clustering output at fourth iteration for case (D)

The KFLANN clustering algorithm converged at the fourth iteration for this dataset. The tolerance setting using the standard deviation may not be the optimal setting however it serves as a good initialization method. With the tolerance tuning to a smaller value for this particular case a better clustering result was obtained.

APPENDIX B – EVALUATION METHOD

B.1 Receiver Operating Characteristics

Receiver Operating Characteristics (*ROC*) is used as the performance evaluation of the tested classification algorithm in this thesis. This method is not sensitive on the skew classes distribution. Its formulation is:

$$E = \frac{1}{K} \sum_{m=1}^K e_m$$

$$e_m = \frac{\text{number of pattern misclassified in class } m}{n_m}$$

Where K is the number of classes, e_m is the classification error for class m and n_m is the number of pattern in class m . Complementing the use of *ROC* measurement is the Area Under Curve (*AUC*):

$$AUC = 1 - E$$

AUC can be viewed as the accuracy measurement. The accuracy in this thesis follows the *AUC* as the accuracy measurement in the percentage format unless mentioned otherwise.

B.2 Normalized Error Ratio

$$R = \frac{1 + E_{original}}{1 + E_{reduced}}$$

$E_{original}$ is the average error on the original feature space while $E_{reduced}$ is the average error on the compressed feature space. Both the denominator and nominator of equation above are adding with 1 to avoid the possible case of zero division. The value of R is within range of [0.5,2.0], if it has the value of 1, can be said that the compressed feature space has the same discriminatory power as the original feature space. When the value R is greater than 1 mean compressed feature space has better discriminatory ability than the original feature space. This is possible since the impact of the “curse of dimensionality”

has been eliminated thru the feature compression. The average error is used here indicates how well compressed feature preserve the original classes content.

B.3 Number of hidden neuron

$$\#hidden = (\#input + \#output) / 2$$

#hidden: Number of hidden node

#input: Number of input node

#output: Number of output node

APPENDIX C – DATASET DESCRIPTION

C.1 UCI Dataset Description

Dataset	Iris																				
Number of instances	120 training 30 testing																				
Number of attributes	4																				
Attributes information	<table><tr><th>Attribute</th><th>Std Dev</th><th>Max</th><th>Min</th></tr><tr><td>Sepal width</td><td>0.2300</td><td>1</td><td>0</td></tr><tr><td>Petal width</td><td>0.1806</td><td>1</td><td>0</td></tr><tr><td>Sepal length</td><td>0.2989</td><td>1</td><td>0</td></tr><tr><td>Petal length</td><td>0.3179</td><td>1</td><td>0</td></tr></table>	Attribute	Std Dev	Max	Min	Sepal width	0.2300	1	0	Petal width	0.1806	1	0	Sepal length	0.2989	1	0	Petal length	0.3179	1	0
Attribute	Std Dev	Max	Min																		
Sepal width	0.2300	1	0																		
Petal width	0.1806	1	0																		
Sepal length	0.2989	1	0																		
Petal length	0.3179	1	0																		
Number of classes	3 (50 each class)																				
Data source	UCI Machine Learning, donor is Michael Marshall, creator is R. A. Fisher.																				
Description	The domain contains 3 classes of Iris flowers, one class is linearly separable from the other two type. While the other 2 flowers are overlapped with each other.																				

Dataset	Thyroid												
Number of instances	140 training 75 testing												
Number of attributes	5												
Attributes information	<table><tr><th>Attribute</th><th>Description</th></tr><tr><td>T3-resin uptake test</td><td>In percentage format</td></tr><tr><td>Total Serum thyroxin</td><td>measured by the isotopic displacement method.</td></tr><tr><td>Total serum triiodothyronine</td><td>measured by radioimmuno assay</td></tr><tr><td>basal thyroid-stimulating hormone (TSH)</td><td>as measured by radioimmuno assay.</td></tr><tr><td>Maximal absolute difference of TSH value</td><td>The reading is obtained after injection of 200 micro grams of thyrotropin-releasing hormone as compared to the basal value.</td></tr></table>	Attribute	Description	T3-resin uptake test	In percentage format	Total Serum thyroxin	measured by the isotopic displacement method.	Total serum triiodothyronine	measured by radioimmuno assay	basal thyroid-stimulating hormone (TSH)	as measured by radioimmuno assay.	Maximal absolute difference of TSH value	The reading is obtained after injection of 200 micro grams of thyrotropin-releasing hormone as compared to the basal value.
Attribute	Description												
T3-resin uptake test	In percentage format												
Total Serum thyroxin	measured by the isotopic displacement method.												
Total serum triiodothyronine	measured by radioimmuno assay												
basal thyroid-stimulating hormone (TSH)	as measured by radioimmuno assay.												
Maximal absolute difference of TSH value	The reading is obtained after injection of 200 micro grams of thyrotropin-releasing hormone as compared to the basal value.												
Number of classes	3 (normal, hyper, hypo)												
Data source	Stefan Aeberhard, Garavan from Department of Computer Science, James Cook University donated this dataset. The original owner of the dataset was belong to Danny Coomans from Department of Mathematic and Statistical, James Cook University.												
Description	The dataset was obtained from 5 laboratory tests. There are used to predict whether a patient's thyroid to the class euthyroidism, hypothyroidism or hyperthyroidism. The diagnosis (the class label) was based on a complete medical record, including anamnesis, scan etc.												

Dataset	Wisconsin Breast Cancer																																								
Number of instances	699																																								
Number of attributes	9																																								
Attributes information	<table><tr><th>Attribute</th><th>Std Dev</th><th>Max</th><th>Min</th></tr><tr><td>Clump Thickness</td><td>0.2823</td><td>1</td><td>0</td></tr><tr><td>Uniformity of Cell Size</td><td>0.3390</td><td>1</td><td>0</td></tr><tr><td>Uniformity of Cell Shape</td><td>0.3302</td><td>1</td><td>0</td></tr><tr><td>Marginal Adhesion</td><td>0.3173</td><td>1</td><td>0</td></tr><tr><td>Single Epithelial Cell Size</td><td>0.2461</td><td>1</td><td>0</td></tr><tr><td>Bare Nuclei</td><td>0.3641</td><td>1</td><td>0</td></tr><tr><td>Bland Chromatin</td><td>0.2710</td><td>1</td><td>0</td></tr><tr><td>Normal Nucleoli</td><td>0.3393</td><td>1</td><td>0</td></tr><tr><td>Mitoses</td><td>0.1905</td><td>1</td><td>0</td></tr></table>	Attribute	Std Dev	Max	Min	Clump Thickness	0.2823	1	0	Uniformity of Cell Size	0.3390	1	0	Uniformity of Cell Shape	0.3302	1	0	Marginal Adhesion	0.3173	1	0	Single Epithelial Cell Size	0.2461	1	0	Bare Nuclei	0.3641	1	0	Bland Chromatin	0.2710	1	0	Normal Nucleoli	0.3393	1	0	Mitoses	0.1905	1	0
Attribute	Std Dev	Max	Min																																						
Clump Thickness	0.2823	1	0																																						
Uniformity of Cell Size	0.3390	1	0																																						
Uniformity of Cell Shape	0.3302	1	0																																						
Marginal Adhesion	0.3173	1	0																																						
Single Epithelial Cell Size	0.2461	1	0																																						
Bare Nuclei	0.3641	1	0																																						
Bland Chromatin	0.2710	1	0																																						
Normal Nucleoli	0.3393	1	0																																						
Mitoses	0.1905	1	0																																						
Number of classes	2 (benign-458, maglinant-241)																																								
Data source	University of Wisconsin Hospital, Madison. It was a compilation by Dr. William H. Wolberg, downloadable from UCI Repository of machine learning databases.																																								

Description The domain is used to distinguish between malignant and benign tumor. The records are taken on the patient with tumor on their breast.

Dataset	Glass																																																												
Number of instances	172																																																												
Number of attributes	13																																																												
Attributes information	<table><tr><th>Attribute</th><th>Min</th><th>Max</th><th>Mean</th><th>Std Dev</th><th>Correlation with class</th></tr><tr><td>RI</td><td>1.5112</td><td>1.5339</td><td>1.5184</td><td>0.0030</td><td>-0.1642</td></tr><tr><td>Na</td><td>10.73</td><td>17.38</td><td>13.4079</td><td>0.8166</td><td>0.5030</td></tr><tr><td>Mg</td><td>0</td><td>4.49</td><td>2.6845</td><td>1.4424</td><td>-0.7447</td></tr><tr><td>Al</td><td>0.29</td><td>3.5</td><td>1.4449</td><td>0.4993</td><td>0.5988</td></tr><tr><td>Si</td><td>69.81</td><td>75.41</td><td>72.6509</td><td>0.7745</td><td>0.1515</td></tr><tr><td>K</td><td>0</td><td>6.21</td><td>0.4971</td><td>0.6522</td><td>-0.0100</td></tr><tr><td>Ca</td><td>5.43</td><td>16.19</td><td>8.9570</td><td>1.4232</td><td>0.0007</td></tr><tr><td>Ba</td><td>0</td><td>3.15</td><td>0.1750</td><td>0.4972</td><td>0.5751</td></tr><tr><td>Fe</td><td>0</td><td>0.51</td><td>0.0570</td><td>0.0974</td><td>-0.1879</td></tr></table>	Attribute	Min	Max	Mean	Std Dev	Correlation with class	RI	1.5112	1.5339	1.5184	0.0030	-0.1642	Na	10.73	17.38	13.4079	0.8166	0.5030	Mg	0	4.49	2.6845	1.4424	-0.7447	Al	0.29	3.5	1.4449	0.4993	0.5988	Si	69.81	75.41	72.6509	0.7745	0.1515	K	0	6.21	0.4971	0.6522	-0.0100	Ca	5.43	16.19	8.9570	1.4232	0.0007	Ba	0	3.15	0.1750	0.4972	0.5751	Fe	0	0.51	0.0570	0.0974	-0.1879
Attribute	Min	Max	Mean	Std Dev	Correlation with class																																																								
RI	1.5112	1.5339	1.5184	0.0030	-0.1642																																																								
Na	10.73	17.38	13.4079	0.8166	0.5030																																																								
Mg	0	4.49	2.6845	1.4424	-0.7447																																																								
Al	0.29	3.5	1.4449	0.4993	0.5988																																																								
Si	69.81	75.41	72.6509	0.7745	0.1515																																																								
K	0	6.21	0.4971	0.6522	-0.0100																																																								
Ca	5.43	16.19	8.9570	1.4232	0.0007																																																								
Ba	0	3.15	0.1750	0.4972	0.5751																																																								
Fe	0	0.51	0.0570	0.0974	-0.1879																																																								
Number of classes	6																																																												
Data source	USA Forensic Science Service																																																												
Description	13 numerical-valued attributes are defined in term of their oxide content.																																																												

Dataset	Wine																																																								
Number of instances	178																																																								
Number of attributes	13																																																								
Attributes information	<table><tr><th>Attribute</th><th>Std Dev</th><th>Max</th><th>Min</th></tr><tr><td>Alcohol</td><td>0.2136</td><td>1</td><td>0</td></tr><tr><td>Malic acid</td><td>0.2207</td><td>1</td><td>0</td></tr><tr><td>Ash</td><td>0.1466</td><td>1</td><td>0</td></tr><tr><td>Alcalinity of ash</td><td>0.1721</td><td>1</td><td>0</td></tr><tr><td>Magnesium</td><td>0.1552</td><td>1</td><td>0</td></tr><tr><td>Total phenols</td><td>0.2158</td><td>1</td><td>0</td></tr><tr><td>Flavanoids</td><td>0.2107</td><td>1</td><td>0</td></tr><tr><td>Nonflavanoid phenols</td><td>0.2348</td><td>1</td><td>0</td></tr><tr><td>Proanthocyanins</td><td>0.1805</td><td>1</td><td>0</td></tr><tr><td>Color intensity</td><td>0.1978</td><td>1</td><td>0</td></tr><tr><td>Hue</td><td>0.1858</td><td>1</td><td>0</td></tr><tr><td>OD280/OD315 of diluted wines</td><td>0.2601</td><td>1</td><td>0</td></tr><tr><td>Proline</td><td>0.2246</td><td>1</td><td>0</td></tr></table>	Attribute	Std Dev	Max	Min	Alcohol	0.2136	1	0	Malic acid	0.2207	1	0	Ash	0.1466	1	0	Alcalinity of ash	0.1721	1	0	Magnesium	0.1552	1	0	Total phenols	0.2158	1	0	Flavanoids	0.2107	1	0	Nonflavanoid phenols	0.2348	1	0	Proanthocyanins	0.1805	1	0	Color intensity	0.1978	1	0	Hue	0.1858	1	0	OD280/OD315 of diluted wines	0.2601	1	0	Proline	0.2246	1	0
Attribute	Std Dev	Max	Min																																																						
Alcohol	0.2136	1	0																																																						
Malic acid	0.2207	1	0																																																						
Ash	0.1466	1	0																																																						
Alcalinity of ash	0.1721	1	0																																																						
Magnesium	0.1552	1	0																																																						
Total phenols	0.2158	1	0																																																						
Flavanoids	0.2107	1	0																																																						
Nonflavanoid phenols	0.2348	1	0																																																						
Proanthocyanins	0.1805	1	0																																																						
Color intensity	0.1978	1	0																																																						
Hue	0.1858	1	0																																																						
OD280/OD315 of diluted wines	0.2601	1	0																																																						
Proline	0.2246	1	0																																																						
Number of classes	3 (class1-59, class2-71, class3-48)																																																								
Data source	The donor of the dataset is Stefan Aeberhard																																																								
Description	The domain is collected as the results of chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.																																																								

Dataset	Heart																																				
Number of instances	150 training 120 testing																																				
Number of attributes	13																																				
Attributes information	<table><tr><th>Attribute</th><th>Categories</th><th>Description</th></tr><tr><td>Age</td><td>-</td><td></td></tr><tr><td>Sex</td><td>2</td><td>Male, female</td></tr><tr><td>Chest pain type</td><td>4</td><td></td></tr><tr><td>Resting blood pressure</td><td>-</td><td>Continuous value</td></tr><tr><td>Serum cholestrol</td><td>-</td><td>Continuous value</td></tr><tr><td>Fasting blood sugar</td><td>-</td><td>Continuous value</td></tr><tr><td>Resting electrocardiographic result</td><td>-</td><td>Continuous value</td></tr><tr><td>Maximum heart rate achieved</td><td>-</td><td>Continuous value</td></tr><tr><td>Exercise induced angina</td><td>-</td><td>Continuous value</td></tr><tr><td>Oldpeak</td><td>-</td><td>Continuous value</td></tr><tr><td>The slope of the peak ST segment</td><td>-</td><td>Continuous value</td></tr></table>	Attribute	Categories	Description	Age	-		Sex	2	Male, female	Chest pain type	4		Resting blood pressure	-	Continuous value	Serum cholestrol	-	Continuous value	Fasting blood sugar	-	Continuous value	Resting electrocardiographic result	-	Continuous value	Maximum heart rate achieved	-	Continuous value	Exercise induced angina	-	Continuous value	Oldpeak	-	Continuous value	The slope of the peak ST segment	-	Continuous value
Attribute	Categories	Description																																			
Age	-																																				
Sex	2	Male, female																																			
Chest pain type	4																																				
Resting blood pressure	-	Continuous value																																			
Serum cholestrol	-	Continuous value																																			
Fasting blood sugar	-	Continuous value																																			
Resting electrocardiographic result	-	Continuous value																																			
Maximum heart rate achieved	-	Continuous value																																			
Exercise induced angina	-	Continuous value																																			
Oldpeak	-	Continuous value																																			
The slope of the peak ST segment	-	Continuous value																																			

	<table><tr><td>Number of major vessels colored by flouroscopy</td><td>-</td><td>Continuous value</td></tr><tr><td>Thal</td><td>3</td><td>Normal, fixed defect, reversible defect</td></tr></table>	Number of major vessels colored by flouroscopy	-	Continuous value	Thal	3	Normal, fixed defect, reversible defect
Number of major vessels colored by flouroscopy	-	Continuous value					
Thal	3	Normal, fixed defect, reversible defect					
Number of classes	2 (presence of heart disease, non-presence of heart disease)						
Data source	Cleveland Clinic Foundation, Hungarian Institute of Cardiology, V.A. Medical Center Long Beach and University Hospital Zurich.						
Description	The objective is to diagnose presence of heart disease of the tested patient.						

Dataset	Ionosphere
Number of instances	351
Number of attributes	34
Attributes information	34 continuous features (The features are the received signals, there were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.)
Number of classes	2 (Good radar and bad radar)
Data source	UCI Machine Learning Repository, donor is Vince Sigillito from Space Physics Laboratory, John Hopkins University.
Description	The dataset is the radar information collected by a system in Goose Bay, Labrador. The system consists of a phased array of 16 high frequency antennas with total transmitted power on the order of 6.4 KW. The targets are free electrons in the ionosphere, good radar returns are showing evidence of some type of structure in the ionosphere while bad radar returns are those that do not (the signal pass through the ionosphere).

Dataset	Dermatology
Number of instances	366
Number of attributes	34
Attributes information	12 Clinical Attributes and histopathological Attributes, both take values 0, 1, 2, 3
Number of classes	6 (psoriasis-112, seboreic dermatitis-61, lichen planus-72, pityriasis rosea-49, cronic dermatitis-52, pityriasis rubra pilaris-20)
Data source	Original owner is Nilsel Liter and H. Altay Guvenir. The donor is H. Altay Guvenir.
Description	The differential diagnosis of 6 types of Eryhemato-Squamous disease in dermatology.

Dataset	Zoo																																																			
Number of instances	82 training, 19 testing																																																			
Number of attributes	17 nominal features																																																			
Attributes information	<table><tr><th>Attribute</th><th>Categories</th><th>Description</th></tr><tr><td>Hair</td><td>2</td><td>Yes, no</td></tr><tr><td>Features</td><td>2</td><td>Yes, no</td></tr><tr><td>Eggs</td><td>2</td><td>Yes, no</td></tr><tr><td>Milk</td><td>2</td><td>Yes, no</td></tr><tr><td>Airbone</td><td>2</td><td>Yes, no</td></tr><tr><td>Aquatic</td><td>2</td><td>Yes, no</td></tr><tr><td>Predator</td><td>2</td><td>Yes, no</td></tr><tr><td>Toothed</td><td>2</td><td>Yes, no</td></tr><tr><td>Backbone</td><td>2</td><td>Yes, no</td></tr><tr><td>Breathes</td><td>2</td><td>Yes, no</td></tr><tr><td>Venomous</td><td>2</td><td>Yes, no</td></tr><tr><td>Fins</td><td>2</td><td>Yes, no</td></tr><tr><td>Legs</td><td>6</td><td>0, 2, 4, , 5, 6, 8</td></tr><tr><td>Tail</td><td>2</td><td>Yes, no</td></tr><tr><td>Domestic</td><td>2</td><td>Yes, no</td></tr><tr><td>Catsize</td><td>2</td><td>Yes, no</td></tr></table>	Attribute	Categories	Description	Hair	2	Yes, no	Features	2	Yes, no	Eggs	2	Yes, no	Milk	2	Yes, no	Airbone	2	Yes, no	Aquatic	2	Yes, no	Predator	2	Yes, no	Toothed	2	Yes, no	Backbone	2	Yes, no	Breathes	2	Yes, no	Venomous	2	Yes, no	Fins	2	Yes, no	Legs	6	0, 2, 4, , 5, 6, 8	Tail	2	Yes, no	Domestic	2	Yes, no	Catsize	2	Yes, no
Attribute	Categories	Description																																																		
Hair	2	Yes, no																																																		
Features	2	Yes, no																																																		
Eggs	2	Yes, no																																																		
Milk	2	Yes, no																																																		
Airbone	2	Yes, no																																																		
Aquatic	2	Yes, no																																																		
Predator	2	Yes, no																																																		
Toothed	2	Yes, no																																																		
Backbone	2	Yes, no																																																		
Breathes	2	Yes, no																																																		
Venomous	2	Yes, no																																																		
Fins	2	Yes, no																																																		
Legs	6	0, 2, 4, , 5, 6, 8																																																		
Tail	2	Yes, no																																																		
Domestic	2	Yes, no																																																		
Catsize	2	Yes, no																																																		
Number of classes	7																																																			
Data source	Dataset was obtained from Richard Forsyth																																																			
Description	Artificial dataset that is used to category different group of animals.																																																			

Dataset	Spambase
Number of instances	4601
Number of attributes	57 continuous
Attributes information	48 continuous real attributes of type word_freq_WORD, 6 continuous real attributes of type char_freq_CHAR, 1 continuous real attribute of type capital_run_length_average, 1 continuous integer attribute of type capital_run_length_longest, 1 continuous integer attribute of type capital_run_length_total.
Number of classes	2 (spam, non-spam)
Data source	UCI Machine Learning, donated by George Forman
Description	The domains recorded characteristic of emails that aim to distinguish between spam email and non-spam email.

Dataset	Sonar
Number of instances	208
Number of attributes	60 continuous attributes
Attributes information	
Number of classes	2 (metal-111, rock-97)
Data source	The data set was contributed to the benchmark collection by Terry Sejnowski, now at the Salk Institute and the University of California at San Deigo.
Description	The data set was developed in collaboration with R. Paul Gorman of Allied-Signal Aerospace Technology Center. The transmitted sonar signal is a frequency-modulated chirp, rising in frequency. The data set contains signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock. Each pattern is a set of 60 numbers in the range 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period of time. The integration aperture for higher frequencies occurs later in time, since these frequencies are transmitted later during the chirp.

Dataset	Waveform
Number of instances	3993 training 1007 testing
Number of attributes	21
Attributes information	Continuous feature between 0 and 6
Number of classes	3
Data source	David Aha is the donor.
Description	The dataset is created from a combination of 2 of 3 base waves, each instance is generated from added noise (mean 0, variance 1) in each attribute.

Dataset	Monks3
Number of instances	Monks3-122
Number of attributes	17
Attributes information	Embedded regulations Monks3 - $a_5 = 3$ and $a_4 = 1$ or ($a_5 \neq 4$ and $a_2 \neq 3$)
Number of classes	2 (0, 1)
Data source	Donor is Sebastian Thrun
Description	The domain is the basic of a first international comparison of learning algorithms.

Dataset	Soya bean
Number of instances	47
Number of attributes	35 nominal features
Attributes information	All attributes take different categories values

Attribute	Categories	Attribute	Categories
Date	8	Leafspot-marg	3
Plant-stand	3	Leafspot-size	3
Precip	3	Leaf-shread	2
Temp	3	Leaf-malf	2
Hail	2	Leaf-mild	3
Crop-hist	4	Stem	2
Area damaged	4	Lodging	2
Severity	3	Stem cankers	4

Seed-tmt	3	Canker lesion	4
Germination	3	Fruiting bodies	2
Plant growth	2	External decay	3
Leaves	2	Mycelium	2
Leafspots-halo	3	Int discolor	3
fruit-pods	3	sclerotia	2
fruit-spots	5	seed	2
mold-growth	2	seed-discolor	2
seed-size	2	shriveling	2
roots	3		

Number of classes 4 (class1-10, class2-10, class3-10, class4-17)
 Data source Donor is Doug Fisher
 Description The dataset contains of attributes that are used to predict the soybean disease.

Dataset Segment
 Number of instances 2310
 Number of attributes 19
 Attributes information

Attributes	Description
region-centroid-col	the column of the center pixel of the region
region-centroid-row	the row of the center pixel of the region
region-pixel-count	the number of pixels in a region = 9
short-line-density-5	the results of a line extraction algorithm that counts how many lines of length 5 (any orientation) with low contrast, less than or equal to 5, go through the region.
short-line-density-2	same as short-line-density-5 but counts lines of high contrast, greater than 5
vedge-mean	measure the contrast of horizontally adjacent pixels in the region. There are 6, the mean and standard deviation are given. This attribute is used as a vertical edge detector
vedge-sd	Standard deviation of the vertical edge detector
hedge-mean	Measures the contrast of vertically adjacent pixels. Used for horizontal line detection
hedge-sd	Standard deviation of the hedge measurement
intensity-mean	the average over the region of $(R + G + B)/3$
rawred-mean	the average over the region of the R value
rawblue-mean	the average over the region of the B value
rawgreen-mean	the average over the region of the G value
exred-mean	measure the excess red: $(2R - (G + B))$
exblue-mean	measure the excess blue: $(2B - (G + R))$
exgreen-mean	measure the excess green: $(2G - (R + B))$
value-mean	3-d nonlinear transformation of RGB. (Algorithm can be found in Foley and VanDam, Fundamentals of Interactive Computer Graphics)
saturation-mean	
Hue-mean	

Number of classes 7
 Classes:
 1 = brickface
 2 = sky
 3 = foliage
 4 = cement
 5 = window
 6 = path
 7 = grass
 Data source The creator of this dataset was Vision Group from University of Massachusetts. Carla Brodley donated the dataset to UCI repository on November 1990.
 Description Dataset contains of instances that were drawn randomly from a database of 7 outdoor images. The images were hand-segmented to create a classification for every pixel.

Dataset	Splice
Number of instances	3190
Number of attributes	60 DNA position
Attributes information	All examples taken from Genbank 64.1 (ftp site: genbank.bio.net). Categories EI and IE include every "split-gene" for primates in Genbank 64.1. For non-splice examples taken from sequences known not to include a splicing site.
Number of classes	3 (Neither-1655, IE-768, EI-767)
Data source	Donor: G. Towell, M. Noordewier, and J. Shavlik,
Description	Splice junctions are points on a DNA sequence at which 'superfluous' DNA is removed during the process of protein creation in higher organisms. The problem posed in this dataset is to recognize, given a sequence of DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out). This problem consists of two subtasks: recognizing exon/intron boundaries (referred to as EI sites), and recognizing intron/exon boundaries (IE sites). (In the biological community, IE borders are referred to as "acceptors" while EI borders are referred to as "donors".)

Dataset	Multi feature
Number of instances	2000 (200 per class)
Number of attributes	649
Attributes information	The dataset contains features of handwritten (0-9) that were extracted from a collection of Dutch utility maps.
Number of classes	10
Data source	The dataset was obtained from Robert P. W. Duin
Description	Digits are represented in terms of Fourier coefficient, profile correlations, Karhunen-Love coefficients, pixels average, Zernike moments and morphological features

Dataset	Arrhythmia
Number of instances	452
Number of attributes	279
Attributes information	The attributes consists of patient's particulars and cardiac reading.
Number of classes	16
Data source	Original source was from H. Altay Guvenir, Bilkent University
Description	The dataset was used to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups.

C.2 Microarray Datasets

Dataset	ALL-AML Leukimia
Number of instances	38 training 34 testing
Number of attributes	7129 numerical
Number of classes	2 (AML, ALL)
Data source	http://www-genome.wi.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43
Description	Training set contains 27 ALL and 11 AML over 7129 probes from 6817 genes. 20 ALL and 14 AML in testing set.

Dataset	Lung Cancer
Number of instances	32 training, 149 testing
Number of attributes	12553
Number of classes	2 (Malignant Pleural Mesothelioma- MPM, Adenocarcinoma – ADCA)
Data source	http://www.chestsurg.org/
Description	Each instance contains of 12533 genes to be classified into 2 classes. There are 16 samples of MPM and 16 ADCA in the training set. The test set contains 15 samples of MPM and 134 ADCA samples.

Dataset	Colon Tumor
Number of instances	62
Number of attributes	2000 numerical
Number of classes	2 (positive, negative)
Data source	http://microarray.princeton.edu/oncology/affydata/index.html
Description	40 tumor biopsies are from tumors (negative) and 22 normal biopsies from healthy part of the colons of the same patient (positive). 2000 out of 6500 genes are selected based on the confidence in the measured expression levels.
Dataset	Ovarian cancer
Number of instances	253
Number of attributes	15154 numerical
Number of classes	2 (cancer, normal)
Data source	http://clinicalproteomics.steem.com/
Description	The objective of the dataset is to identify proteomic patterns in serum that distinguish ovarian cancer from non-cancer. The proteomic spectra were generated by mass spectroscopy and the data set provided here is 6-19-02, which includes 91 controls (Normal) and 162 ovarian cancers. The raw spectral data of each sample contains the relative amplitude of the intensity at each molecular mass / charge (M/Z) identity. There are total 15154 M/Z identities. The intensity values were normalized according to the formula: $NV = (V - \text{Min}) / (\text{Max} - \text{Min})$, where NV is the normalized value, V the raw value, Min the minimum intensity and Max the maximum intensity. The normalization is done over all the 253 samples for all 15154 M/Z identities
Dataset	Prostate Cancer
Number of instances	102 Training, 34 Testing
Number of attributes	12600 numerical
Number of classes	2 (Tumor, Normal)
Data source	http://www-genome.wi.mit.edu/mpr/prostate http://carrier.gnf.org/welsh/prostate/
Description	Classification of tumor from normal cell of prostate.

C.3 Neural Information Processing System (NIPS 2003) Datasets

Dataset	Madelon
Number of instances	2000 training 600 testing
Number of attributes	500 (20 real features 480 random probes)
Attributes information	All features are integer
Number of classes	2
Data source	NIPS 2003 Dataset
Description	The goal is to classify random synthetic data. Each class contains a number of Gaussian clusters. $N(1,0)$ is used to determine number of useful features examples of independent features. Some covariance is added by multiplying with random matrix (uniformly distributed random numbers between -1 and 1). Redundant features are added through multiplication of useful features by a random matrix. Random probes are added using $N(1,0)$ together with random noise added according to $N(0,0.1)$.
Dataset	Arcane
Number of instances	100 training 100 testing
Number of attributes	10000 (7000 real features 3000 random probes)
Attributes information	All features are integer quantized on 1000 levels.
Number of classes	2
Data source	NIPS 2003 Dataset, original from National Cancer Institute (NCI) and Eastern Virginia Medical School (EVMS).
Description	The dataset is used to classify cancer versus normal patterns from mass-spectrometric data that was obtained using SELDI technique. The data samples included patients with cancer (prostate or ovarian) and those healthy

and control patients.

Dataset	Gisette
Number of instances	6000 training 6500 testing
Number of attributes	5000 (2500 real features, 2500 random probes)
Attributes information	All features are integer quantized on 1000 levels, approximate 313% of the data are non zero, thus it is a sparse data.
Number of classes	2
Data source	NIPS 2003 Dataset, origin from the MNIST by Yann LeCun of the NEC Research Institute at http://yann.lecun.com/exdb/mnist/
Description	The goal of the dataset is to distinguish between confusable handwritten digits (4 and 9)

C.4 Artificial Dataset

Dataset	Dimension	Number of Data Pattern	Number of Classes
AD6	10	1000	2
AD7	10	1000	2
AD8	10	1000	2
AD9	10	1000	2
AD10	10	1000	2
AD11	10	1000	2
AD12	10	1000	2

APPENDIX D – PUBLICATIONS

1. The K-means fast learning artificial neural network (KFLANN), an atomic model for hierarchical network architectures, 9th International Conference On Cognitive And Neural System, 2005.
2. The hierarchical fast learning artificial neural network: A framework for scalable learning, 9th International Conference On Cognitive And Neural System, 2005.
3. Centroid Stability with K-Means Fast Learning Artificial Neural Networks, International Joint Conference on Neural Network 2003 (IJCNN 2003).
4. Liquid Drop Photonic Signal Analysis Using Fast Learning Artificial Neural Network, 4th International Conference On Information Communications and Signal Processing (ICICS 2003).
5. Hierarchical Fast Learning Artificial Neural Network, International Joint Conference on Neural Network 2005 (IJCNN 2005).
6. A Novel Hybrid Learning Scheme For Pattern Recognition, International Joint Conference on Neural Network 2005 (IJCNN 2005).
7. Columnar Projection Development of Hierarchical Neural Networks using the K-Means Fast Learning Artificial Neural Network, IEEE Transaction on Neural Network 2005 (under review).
8. The Hierarchical Fast Learning Artificial Neural Network (HieFLANN) – An Autonomous Platform for Hierarchical Neural Network Construction, IEEE Transaction on Neural Network 2005 (under review).