STUDIES ON SUPERVISED CLASSIFICATION ALGORITHMS BASED ON DATASET TRANSFORMATION FOR MONITORING NUCLEAR POWER PLANT EVENTS

By

MANAS RANJAN PRUSTY

(Enrollment No: ENGG02201104037)

INDIRA GANDHI CENTRE FOR ATOMIC RESEARCH
KALPAKKAM

A thesis submitted to the

Board of Studies in Engineering Sciences

In partial fulfillment of requirements

For the Degree of

DOCTOR OF PHILOSOPHY

of

HOMI BHABHA NATIONAL INSTITUTE

JUNE, 2016

HOMI BHABHA NATIONAL INSTITUTE

Recommendations of the Viva Voce Board

As members of the Viva Voce Board, we certify that we have read the dissertation prepared by Mr. Manas Ranjan Prusty entitled "Studies on supervised classification algorithms based on dataset transformation for monitoring nuclear power plant events" and recommend that it may be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Chairman: **Dr. B. K. Panigrahi** Date: 07/6/2017

Guide/Convener: **Dr. K. Velusamy** Date: 7/6/2017

External Examiner: **Dr. M Narasimha Murty** Date: 7-6-2017

Member 1: **Dr. Anish Kumar** Date: 7/6/2017

Member 2: **Dr. V. S. Srinivasan** Date: 7/6/2017

Technology Advisor: **Smt. T. Jayanthi** Date: 7/6/2017

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertations to HBNI.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it may be accepted as fulfilling the dissertation requirement.

Date: 7/6/2017

Place: Indira Gandhi Centre for Atomic Research (IGCAR)
Kalpakkam

Dr. K. Velusamy
(Guide)

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at Homi Bhabha National Institute (HBNI) and is deposited in the library to be made available to borrowers under rules of the HBNI.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the Competent Authority of HBNI when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

Date : **(MANAS RANJAN PRUSTY)**

Place : Kalpakkam

DECLARATION

I, hereby declare that the investigation presented in the thesis has been carried out by me. The work is original and has not been submitted earlier as a whole or in part for a degree / diploma at this or any other Institution / University.

Date : **(MANAS RANJAN PRUSTY)**

Place : Kalpakkam

LIST OF PUBLICATIONS ARISING FROM THE THESIS

JOURNALS

1) **M. R. Prusty**, T. Jayanthi, J. Chakraborty, H. Seetha, K. Velusamy, "Performance analysis of fuzzy rule based classification system for transient identification in nuclear power plant", Annals of Nuclear Energy, 76, 2015, 63–74.

2) **M. R. Prusty**, T. Jayanthi, K. Velusamy, "Feasibility of ANFIS towards multiclass event classification in PFBR considering dimensionality reduction using PCA", Annals of Nuclear Energy, 99, 2017, 311-320.

3) **M. R. Prusty**, T. Jayanthi, K. Velusamy, "Weighted-SMOTE: A modification to SMOTE", Knowledge and Information Systems, (Revised version submitted).

4) **M. R. Prusty**, T. Jayanthi, K. Velusamy, "Training dataset reduction for classifier networks: A novel approach", Pattern Recognition Letters, (Under review).

CONFERENCES

1) **M. R. Prusty**, J. Chakraborty, H. Seetha, T. Jayanthi, K. Velusamy, " Fuzzy logic based transient identification system for operator guidance using prototype fast breeder reactor operator training simulator", Proc. of IEEE Int. Advance Computing Conference, 21st – 22nd February, 2014, ITM University, Gurgaon, India.

2) **M. R. Prusty**, J. Chakraborty, T. Jayanthi, K. Velusamy, "Performance Comparison of Supervised Machine Learning Algorithms for Multiclass Transient Classification in a Nuclear Power Plant", Proc. 5th Int. Conf. of Swarm, Evolutionary and Memetic Computing & Fuzzy and Neural Computing, 18th – 20th December, 2014, SOA University, Bhubaneswar, India.

“I dedicate this thesis to my family for their love, care and support”

ACKNOWLEDGEMENT

At the outset, I would like to thank my research supervisor, Dr. K. Velusamy, Head, Mechanics and Hydraulics Division, Reactor Design Group and my technology advisor, Smt. T. Jayanthi, Head, Simulator Section, Electronics Instrumentation and Radiological Safety Group, IGCAR for their invariable motivation, guidance, encouragement and fortitude during the course of my research tenure at IGCAR. I would like to express my sincere gratitude to both of them for their supervision in many aspects including literature survey, investigational work, preparation of research articles, completion of thesis and exhaustive deliberations regarding supervised classification algorithms.

I convey my heartiest appreciation to my doctoral committee Chairman Prof. B. K. Panigrahi and Committee members Prof. Anish Kumar and Prof. V. S. Srinivasan for their support and encouragement during my research work.

My sincere gratitude goes to Dr. S. A. V. Satya Murty, Director, IGCAR and Dr. P. R. Vasudeva Rao and Shri S. C. Chetal, former Directors, IGCAR for providing excellent environment to carry out research work.

My sincere thanks are owed to Mrs. H. Seetha for giving me the basic idea about Fast Breeder Reactor Systems and also for her guidance. I am also indebt to Mr. Jaideep Chakraborty for his knowledge sharing on artificial intelligence and understanding the fundamental ideas of it.

I express my sincere acknowledgement to Ms. Prema Soundrarajan, Mrs. Deepika Vinod and Mr. T. S. Shri Krishnan for their moral support and going through my thesis and giving me valuable suggestions during my entire PhD work.

It is my pleasure to thank Mr. Narayanan, Mrs. Bindu, Mrs. Jasmine, Mr. Subba Raju, Mrs. Priyanka, Mr. Subrahmanyam, Mr. Nakul, Mr. Raghava and Mrs. Uma Maheswari of Simulator Section for their support and discussions throughout my tenure. I would also like to thank Vinita, Molly, Mr. Parameswaran, Prashant, Pradeep, Vikas and all members of Computer Division for creating a homely environment at the work place during this period.

I am grateful for the financial support provided by research fellowship scheme from Indira Gandhi Centre for Atomic Research, Department of Atomic Energy for the duration of the work.

I am filled with extreme joy to thank my friends, Subrata, Karthickeyan, Samba Siva, Thangam, Sanjay, my cricket team mates, my badminton team mates and all friends at Research Scholar's Enclave for making life so lively and happening during my stay.

Finally, I would like to thank my mother Smt. Santilata Prusty, my father Shri Upendra Kumar Prusty and my wife Mrs. Itishree Ghatuari who have been the source of constant inspiration and motivation for me throughout my life. My heart is filled with diligent gratitude to my in-laws and my family members who have created an ideal environment for me to smoothly handle my career and bring the best out of me in all the endeavors in my life.

Thank you everyone

Manas Ranjan Prusty

TABLE OF CONTENTS

Synopsis……i

List of Figures……..v

List of Tables……...ix

List of Abbreviations……...x

1 INTRODUCTION 1

1.1 Introduction 1

1.2 Nuclear Power Plant 2

1.3 Event Classification 5

1.4 Problem Statement 8

1.5 Objective and Scope of work 8

1.6 Thesis Structure 10

2 BACKGROUND INFORMATION 13

2.1 Fuzzy logic and fuzzy inference system 13

2.1.1 Membership function (MF) 15

2.1.2 Mamdani-type FRBS 15

2.1.3 Takagi-Sugeno-Kang (TSK) type FRBS 20

2.2 Supervised Machine Learning Algorithms 22

2.2.1 k-nearest neighbor 23

2.2.2 Support vector machine 24

2.2.3 Artificial neural network 26

2.3 Performance metrics 34

3 FUZZY RULE BASED CLASSIFICATION SYSTEM FOR TRANSIENT IDENTIFICATION 39

3.1 Introduction 39

3.2 FRBCS for transient identification 41

3.3 Phase-1 of FRBCS 42

3.3.1 FRBCS for Phase–1 using θ_{RI} and $\Delta\ \theta_{RI}$ (Phase–1a) 44

3.3.1.1 Membership functions for Phase-1a 44
3.3.1.2 Rule Base for Phase-1a 46
3.3.1.3 Output for Phase-1a 47
3.3.1.4 Response time for Phase-1a 48
3.3.2 FRBCS for Phase–1 using DL and ΔDL (Phase–1b) 49
3.3.2.1 Membership functions for Phase-1b 50
3.3.2.2 Rule base for Phase-1b 51
3.3.2.3 Output for Phase-1b 52
3.3.2.4 Response time for Phase-1b 53
3.3.3 Comparison between Phase-1a and Phase-1b 54
3.4 Phase-2 of FRBCS 54
3.4.1 FRBCS for Phase–2 using DL and ΔDL (Phase–2a) 56
3.4.1.1 Membership functions for Phase-2a 56
3.4.1.2 Rule base for Phase-2a 58
3.4.1.3 Output for Phase-2a 59
3.4.1.4 Response time for Phase-2a 60
3.4.2 FRBCS for Phase–2 using DL and CP (Phase–2b) 61
3.4.2.2 Rule base for Phase-2b 62
3.4.2.3 Output for Phase-2b 63
3.4.2.4 Response time for Phase-2b 64
3.4.3 FRBCS for Phase–2 using DL, ΔDL and CP (Phase–2c) 65
3.4.3.1 Membership functions for Phase-2c 65
3.4.3.2 Rule base for Phase-2c 67
3.4.3.3 Output for Phase-2c 67
3.4.3.4 Response time for Phase-2c 68
3.4.4 Comparison between response times of the three FRBCS in Phase-2 69
3.5 Summary 70

4 DEVELOPMENT OF TRAINING DATASET REDUCTION FOR TRANSIENT IDENTIFICATION 72

4.1 Introduction 72

4.2 Concept of Training Dataset Reduction 74

4.2.1 Normalization 75

4.2.2 Steps involved in TDR 76

4.2.3 Mean – αStandard deviation (M-αSD) method 78

4.2.4 Area selection Method 79

4.2.4.1 Analytical Integration 80

4.2.4.2 Numerical Integration 81

4.3 Procedure 82

4.4 Observation 84

4.5 Transient classification in a nuclear power plant using TDR approach 97

4.6 Summary 101

5 DEVELOPMENT OF DIMENSIONALITY REDUCTION BASED CLASSIFIER FOR MALFUNCTION IDENTIFICATION 103

5.1 Introduction 103

5.2 Adaptive network based fuzzy inference system 105

5.2.1 Layer 1 107

5.2.2 Layer 2 108

5.2.3 Layer 3 108

5.2.4 Layer 4 109

5.2.5 Layer 5 109

5.3 Principal Component Analysis 110

5.4 Experimental Procedure 114

5.4.1 Secondary Sodium Pump (SSP) 115

5.4.2 Boiler Feed Pump (BFP) 116

5.4.3 Events and features considered 116

5.4.4 Classifier 119

5.4.5 Procedure 120

5.5 Observation 122

5.5.1 Scree Test 122
5.5.2 Testing Phase of CASE 1: Using multiple classifiers - MANFIS 123
5.5.3 Test Phase of CASE 2: Using one classifier 127
5.6 Results 128
5.7 Summary 131

6 DEVELOPMENT OF WEIGHTED SMOTE ALGORITHM FOR IMBALANCED DATASET 132
6.1 Introduction 132
6.2 Synthetic minority over-sampling technique (SMOTE) 135
6.3 Weighted Synthetic minority over-sampling technique (WSMOTE) 136
6.4 Experimental Procedure 138
6.4.1 Experiment 138
6.4.1.1 Datasets 139
6.4.1.2 Classifier Network 140
6.4.2 Procedure 141
6.5 Observation 142
6.6 Performance of WSMOTE and SMOTE for imbalanced dataset classification in a SFR 149
6.7 Summary 152

7 DEVELOPMENT OF GRAPHICAL USER INTERFACE FOR PLANT MONITORING 154
7.1 Introduction 154
7.2 Salient Features of GUI 155
7.3 GUI for fuzzy rule based classification system 156
7.4 GUI for machine learning based classification 157
7.5 Modified GUI for machine learning based classification 160
7.6 Summary 162

8 SUMMARY AND SCOPE FOR FUTURE WORK OF THE THESIS 164
8.1 Summary of the thesis 164
8.2 Scope for future work of the thesis 168

REFERENCES..... **170**

SYNOPSIS

A nuclear power plants (NPP) consists of a large number of subsystems having varied response times. The concept of ‘redundancy and independency’ is followed seriously in the design of the NPP to ensure safety of the plant, public and environment from possible effects of a severe accident. Safety can be ensured by suitable design, proper monitoring of the current status and appropriate reaction to any adverse situation in the plant. The main control room (MCR) in a nuclear power plant (NPP) is assigned the primary responsibility of monitoring and controlling the current status in various subsystems of the plant. Any major imbalance in the plant caused due to any design basis event needs to be reported in the MCR at the earliest. Further, proper handling of such events leads to customary functioning of the plant again. Improper handling or delayed actions to these design basis events can affect the plant adversely. Quick and unambiguous identification of such events and proper handling of the events are the essential plant protection operations. The task of identifying the events is otherwise known as classification problem. One of the ways to deal with the classification problem is using supervised machine learning. Supervised machine learning is a procedure used to train a network which calculates the output for a set of labeled inputs. Here, the labeled inputs are known as training data along with its true output or class or label or event during the learning phase of the network. Another way to counter the classification problems is by the usage of soft computing techniques. A network or system which is exclusively trained to combat classification problems is known as a classifier. The main objective of a classifier in a NPP is to achieve the maximum possible accuracy. The events which have been considered as class or labels in the thesis are

mostly some of the transients in the steam water system along with some malfunctions in typical 500MWe pool type sodium cooled fast reactor (SFR).

Fuzzy rule based classification system (FRBCS) is a method to tackle classification problems. The combination of a well crafted rule base along with a set of properly assigned input and output membership functions helps in classifying the dataset using FRBCS. The present research work focuses on the performance analysis of FRBCS for transient identification in a SFR along with the importance of feature selection to have higher interpretability and acceptable accuracy. It also investigates the usefulness of monitoring the output of a classifier for a certain period of time instead of instantaneous conclusion, in case of online event identification in order to ascertain the occurrence of a particular transient.

A novel approach to reduce the number of training samples has been developed which is known as training dataset reduction (TDR) approach. In TDR approach, some of the training samples are discarded based on the Euclidean distance calculated using a portion of test dataset. A cut-off Euclidean distance is calculated using TDR approach which creates a hypothetical boundary and selects the samples which have lesser Euclidean distance than it. The resulting reduced training dataset is fed as input to a classifier which gets trained using a supervised machine learning algorithm. The performance of the TDR approach on some of the real world datasets and the feasibility of this approach in classifying some of the transients in SFR has also been studied.

The preprocessing of the training dataset in order to filter out necessary information from the huge bank of data provides a greater impact on the training process and eventually helps in improving the classification accuracy. One way of preprocessing the training dataset to reduce the quantum of data is carried out using various dimensionality reduction techniques such as

principal component analysis (PCA). PCA dumps the lesser important features from the calculated principal components thereby reducing the number of columns in the training dataset. Further, this reduced training dataset is fed as input to a classifier that gets trained. One of the extensively used classifier is an adaptive neuro-fuzzy inference system (ANFIS) which has the advantage of both neural networks and fuzzy inference system. In ANFIS, the neural network concept is used to tune the fuzzy membership function. This thesis emphasizes on the research undertaken on the feasibility of usage of PCA based ANFIS for multiclass event classification in SFR considering dimensionality reduction.

Another common issue associated with any classification dataset is the problem of imbalanced dataset. Traditionally, the classification accuracy is biased towards majority class thereby neglecting the minority class. One of the solutions to such problem is oversampling of the minority class samples known as synthetic minority oversampling technique (SMOTE), wherein, each minority sample generates equal number of synthetic data in order to make the dataset balanced. In this thesis, a modification to SMOTE which solves the imbalanced dataset problem, termed as weighted SMOTE is studied. In this algorithm, instead of oversampling each minority sample with same amount of synthetic data, different weights are assigned to each minority sample. Based on the weights assigned, the generation of the amount of synthetic data varies for each minority sample. Finally, the performance of the weighted SMOTE on various real world imbalanced datasets is compared with the traditional SMOTE.

As the MCR in a NPP is studded with many important consoles and panels consisting of numerous alarms, displays, hooters, etc, a proper graphical user interface (GUI) to display any result is extremely crucial. A user friendly GUI with all the necessary information about the condition of the plant is an asset for the operators in the MCR. Information overloading on the

operators can cause panic stricken decision making which eventually may lead to an unsafe circumstance. Hence, the main objective of a GUI must be to display the most obligatory information at any instance, avoiding information overloading on the operator. Instead of displaying numerical data, the demonstration of such information are also made in a more informative manner such as graphs, pie charts, bar graphs, etc, based on requirement and feasibility. This approach develops an improved decision making ability of the operator as the information is conveyed to them in a lucid manner.

The present thesis attempts to understand the various supervised machine learning algorithms for classification problems and their implementation and feasibility with respect to various events in a NPP. The feasibility deals with mostly focusing on the classification accuracy of the classifier. The thesis also attempts on analyzing various aspects of the training data and processing the same in order to achieve better classification accuracy. Further, the GUIs developed and mentioned in the thesis provide a better idea on various information by which the operator can be benefited during an emergency situation.

LIST OF FIGURES

Figure 1.1 : Flow sheet of a 500MWe Fast Breeder Reactor 5

Figure 2.1: Flow sheet of a fuzzy inference system 15

Figure 2.2: Block diagram of Mamdani-type fuzzy rule based system 16

Figure 2.3: Diagrammatic representation of the membership functions operation in Mamdani type FRBS 17

Figure 2.4 : Block diagram of Takagi-Sugeno-Kang-type fuzzy rule based system 21

Figure 2.5: Classification using kNN Algorithm 23

Figure 2.6: Basic decision boundary learned by support vector machine 26

Figure 2.7: Architecture of a basic artificial neural network 27

Figure 2.8: A basic artificial neuron model 27

Figure 2.9: Commonly used activation functions 29

Figure 2.10: Architecture of a multilayer perceptron 32

Figure 2.11: Architecture of a radial basis function network 33

Figure 2.12: Basic representation of a confusion matrix for a binary classifier 35

Figure 2.13: Diagram showing a receiver operating characteristics 38

Figure 3.1: Time series pattern of θ_{RI} for the considered transients 43

Figure 3.2: Time series pattern of DL for the considered transients 44

Figure 3.3: Membership functions of FRBCS for Phase-1a 46

Figure 3.4: Defuzzified output of FRBCS for Phase-1a 48

Figure 3.5: Membership functions of FRBCS for Phase-1b 51

Figure 3.6: Defuzzified output of FRBCS for Phase-1b 53

Figure 3.7: Time series pattern of DL for the considered transients 55

Figure 3.8: Time series pattern of CP for the considered transients 56

Figure 3.9 : Defuzzified output of FRBCS for Phase-1b 58

Figure 3.10: Defuzzified output of FRBCS for Phase-2a 60

Figure 3.11: Membership function of FRBCD for Phase-2b 62

Figure 3.12: Defuzzified output of FRBCS for Phase-2b 64

Figure 3.13: Membership functions of FRBCS for Phase-2c 66

Figure 3.14: Defuzzified output of FRBCS for Phase-2c 68

Figure 4.1: Basic block diagram of TDR approach 77

Figure 4.2: Pictorial representation of selecting $ED_{cut\text{-}off}$ in TDR 78

Figure 4.3: Normal distribution Curve 79

Figure 4.4 : Area selection method for Normal or Gaussian distribution Curve 82

Figure 4.5: Block diagram of training of a benchmark dataset in a k-fold cross validation manner 84

Figure 4.6: Performance of Iris dataset using M-αSD based TDR approach 87

Figure 4.7: Performance of Wine dataset using M-αSD based TDR approach 88

Figure 4.8: Performance of Glass dataset using M-αSD based TDR approach 88

Figure 4.9: Performance of WDBC dataset using M-αSD based TDR approach 89

Figure 4.10: Performance of Balanced Scale dataset using M-αSD based TDR approach 89

Figure 4.11:Performance of BCW dataset using M-αSD based TDR approach 90

Figure 4.12: Performance of Wine Quality Red dataset using M-αSD based TDR approach 90

Figure 4.13: Performance of Wine Quality White dataset using M-αSD based TDR approach .. 91

Figure 4.14: Performance of Iris dataset using area selection based TDR approach 92

Figure 4.15: Performance of Wine dataset using area selection based TDR approach 93
Figure 4.16: Performance of Glass dataset using area selection based TDR approach 93
Figure 4.17: Performance of WBDC dataset using area selection based TDR approach 94
Figure 4.18: Performance of Balanced Scale dataset using area selection based TDR approach 94
Figure 4.19: Performance of BCW dataset using area selection based TDR approach 95
Figure 4.20: Performance of Wine Quality Red dataset using area selection based TDR approach 95
Figure 4.21: Performance of Wine Quality White dataset using area selection based TDR approach 96
Figure 4.22: Time series pattern of DL for the considered transients 98
Figure 4.23: Time series pattern of ΔDL for the considered transients 98
Figure 4.24: Performance of the BFP Trip classifier using M-αSD based TDR approach 99
Figure 4.25: Performance of the CEP Trip classifier using M-αSD based TDR approach 99
Figure 4.26: Performance of the BFP trip classifier using area selection based TDR approach 100
Figure 4.27: Performance of the CEP Trip classifier using area selection based TDR approach 101
Figure 5.1: Typical architecture of an ANFIS network 107
Figure 5.2: Block diagram of the flowchart of PFBR 115
Figure 5.3: Occurrence of the considered seven variables in the plant schematic 118
Figure 5.4: Block diagram of the training process in case-1 120
Figure 5.5: Block diagram of the testing process in case-1 120
Figure 5.6: Scree test of the training data set 123
Figure 5.7: Histogram representation of AUC_{final} for test data set in ANFIS-1 of case-1 126

Figure 5.8: Histogram representation of AUC_{final} for test data set in ANFIS-2 of case-1 127

Figure 5.9: Histogram representation of AUC_{final} for test data set in ANFIS of case-2 128

Figure 5.10: ROC curve of PCA4 in ANFIS-1 of case-1 for 200 epochs 130

Figure 5.11: ROC curve of PCA4 in ANFIS-2 of case-1 for 200 epochs 130

Figure 6.1: Block diagram of SMOTE Generation matrix 136

Figure 6.2: Preprocessing procedure of a dataset 142

Figure 6.3: Comparison of Recall for Ecoli dataset..... 144

Figure 6.4: Comparison of F-Measure for Ecoli dataset..... 144

Figure 6.5: Comparison of Recall for Abalone dataset..... 145

Figure 6.6: Comparison of F-Measure for Abalone dataset 145

Figure 6.7: Comparison of Recall for Wine Quality dataset 146

Figure 6.8: Comparison of F-Measure for Wine Quality dataset 146

Figure 6.9: Comparison of Recall for Yeast dataset..... 147

Figure 6.10: Comparison of F-Measure for Yeast dataset..... 147

Figure 6.11: Comparison of Recall for Mammography dataset 148

Figure 6.12: Comparison of F-Measure for Mammography dataset 148

Figure 6.13: Comparison of Recall for feed water speed dataset 150

Figure 6.14: Comparison of F-Measure for feed water speed dataset 150

Figure 6.15: Comparison of Recall for SSP speed dataset 151

Figure 6.16: Comparison of F-Measure for SSP speed dataset 152

Figure 7.1: GUI-FRBCS classification tool for full power state 157

Figure 7.2: GUI-FRBCS classification tool for transients..... 157

Figure 7.3: GUI-MLC-PCA classification tool 160

Figure 7.4: GUI-ML-TDR-PCA-1 classification tool 161

Figure 7.5: GUI-ML-TDR-PCA-2 classification tool 162

LIST OF TABLES

Table 3.1: Rule base of FRBCS for Phase –1a 47

Table 3.2: Rule base of FRBCS for Phase-1b..... 52

Table 3.3: Response time of the two FRBCS in Phase-1 54

Table 3.4: Rule base of FRBCS for Phase-2a..... 59

Table 3.5: Rule base of FRBCS for Phase-2b..... 63

Table 3.6: Rule base of FRBCS for Phase-2c..... 67

Table 3.7: Response time of the three FRBCS in Phase-2 70

Table 4.1: Description of Real world datasets 84

Table 4.2: Description of markers in graph 85

Table 5.1: Events in PFBR considered for classification 118

Table 5.2: AUC_{final} for test data set in ANFIS-1 of case-1 125

Table 5.3: AUC_{final} for test data set in ANFIS-2 of case-1 127

Table 5.4: AUC_{final} for test data set in ANFIS of case-2 128

Table 5.5: Comparison of AUC_{final} of PCA4 from Tables 5.3 and 5.4..... 129

Table 6.1: Example using Weighted SMOTE 138

Table 6.2: Dataset Distribution..... 140

Table 7.1: Mapping of defuzzified value with transients 156

Table 7.2: Full form of the Backpropagation algorithms 159

LIST OF ABBREVIATIONS

ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
AUC	Area Under ROC Curve
BFP	Boiler Feed Pump
CEP	Condensate Extraction Pump
CP	Condensate Pressure
DBE	Designed Basis Event
DL	Deaerator Level
ED	Euclidean Distance
FBR	Fast Breeder Reactor
FIS	Fuzzy Inference System
FRBCS	Fuzzy Rule Based Classification System
FRBS	Fuzzy Rule Based System
GUI	Graphical User Interface
IHX	Intermediate Heat Exchanger
kNN	k- Nearest Neighbor
MANFIS	Multiple Adaptive Neuro-Fuzzy Inference System
MCR	Main Control Room
MLP	Multilayer Perceptron
MOX	Mixed Oxide
NPP	Nuclear Power Plant
OGDHR	Operation Grade Decay Heat Removal System
OTS	Operator Training Simulator
PCA	Principal Component Analysis
PFBR	Prototype Fast Breeder Reactor
RBF	Radial Basis Function
ROC	Receiver Operating Characteristics
SCR	Sodium Cooled Reactor

SG	Steam Generator
SGDHR	Safety Grade Decay Heat Removal System
SML	Supervised Machine Learning
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
SWS	Steam Water System
TDR	Training Dataset Reduction
θ_{RI}	Reactor Inlet Temperature

1

INTRODUCTION

The present chapter introduces the classification problems and highlights the relevance of the study of classification of events in nuclear power plant. A brief description on nuclear power plant emphasizing on medium sized Fast Breeder Reactor and the importance of event classification is outlined in the present chapter. It delineates some of the major soft computing and machine learning algorithms also.

1.1 Introduction

Classification of events accurately in a small scale or a large scale industry has a significant impact on the overall outcome of the industry. The malfunctioning of any component or the occurrence of failure of any component must be monitored and controlled in order to avoid any imbalance in the industry. If this minor imbalance is not noticed and not treated on time then it might lead to major disaster affecting human beings along with the environment. This particular kind of industry or system is known as safety critical system. An example of a safety critical system is a nuclear power plant (NPP) which is a complex engineering. It consists of a huge number of complex systems integrated together in a systematic manner along with their control and support systems made up of several components. Hence, the safety of the plant largely depends on proper classification of any design basis events at the earliest so that proper steps can be taken to overcome any catastrophic situation.

The monitoring of the current status of the plant is brought to a higher confidence level if there is a suitable equilibrium between the physical models and empirical models. The physical model and empirical model works on deterministic and probabilistic techniques respectively to produce output from a model. The status at a particular instance becomes more obvious and convincing if both these models generate identical results and thereby validate each other. Hence, the final status is a result of an integrated decision making ability of both these models. Generally, probabilistic models are bit faster than the deterministic models. This aides the operator prepare for the upcoming scenario before the integrated results. The present thesis focuses on discussing a variety of empirical methods and its practicability that would assist in identifying various events of a NPP.

1.2 Nuclear Power Plant

For a country like India with the second largest population in the world, sufficing the energy requirement for power generation has always been a challenging mission. The primary source of energy generation in India is coal and based on the amount of carbon dioxide emission, it has a callous impact on the environment. The nuclear power generation has played a noteworthy role in counteracting these issues. The three stage nuclear power program formulated by Dr. Homi Bhabha aims at efficiently utilizing the abundant thorium resources in India to pacify the increasing energy demand. It is a three generation of nuclear reactors using three different fuels. In the first stage, natural or enriched uranium is used as nuclear fuels for thermal reactors. The Plutonium-239 reprocessed from the depleted fuel for the first stage reactors is used as fuel along with Uranium-238 and Thorium-232 blankets in the second stage breeder reactors such as Prototype Fast Breeder Reactor (PFBR). The excess Plutonium-239 produced in these reactors is used to fuel the new breeder reactors. The Uranium-233 produced in the second

stage along with Thorium-232 blankets is used as fuel in the third stage reactors to complete the Uranium-Thorium cycle.

As every NPP is a safety critical process with safety being the major goal along with power production, utmost importance and obligatory measures have been in use to accomplish both [1, 2]. One of the ways of achieving safety is by continuously monitoring the current status of the plant and thereby taking suitable measures on being informed about the occurrence of any event which might cause imbalance in the plant. This can be put into action in a constructive manner only if the event which has occurred is correctly diagnosed. An erroneous detection of the incurred event may challenge plant control and safety system eventually leading to severe accident. Hence, event classification plays an imperative responsibility in attaining the perfect balance of a plant as huge as the PFBR.

PFBR is a 500MWe NPP which is in an advanced stage of commissioning at Kalpakkam, India. It is a pool type, MOX (Mixed Oxide of Plutonium and Uranium) fuel Fast Breeder Reactor with sodium as coolant. A pool type reactor has a superior safety features which enables its selection over loop type reactor. The advantages of MOX fuel are safe operation to high burnup, ease of fabrication and proven reprocessing. The reactor core is a compact core containing 181 fuel subassemblies and 12 absorber rods. The absorber rods are divided as control and safety rods and diverse safety rods with 9 and 3 rods respectively arranged in two rings. The heat transport system consists of primary sodium circuit, secondary sodium circuit and steam water system. The thermal energy generated in the reactor core is transferred through the primary and secondary sodium circuits and ends up in the steam generator producing steam. The steam water system produces superheated steam which drives the Turbo Generator to produce electric power. In case of reactor shutdown, the decay heat retained due to fission reaction is removed by

two decay heat removal systems – operation grade decay heat removal system (OGDHR) which is an active method and safety grade decay heat removal system (SGDHR) which is a passive method. The complete flow sheet of a 500MWe FBR containing the major components is shown in Fig. 1.1. To ensure safety, a defense-in-depth philosophy, consisting of three levels of safety, i.e., design with adequate safety margin, early detection of abnormal events to prevent accidents and mitigation of consequences of accidents, if any, is adopted [3].

PFBR has many complex systems which need proper training to handle. To provide extensive and elaborate training to the operator about the different plant operations and conditions, a full scope replica type operator training simulator (OTS) has been developed [4]. Efficient plant operation depends upon two main factors, i.e., well defined operating procedures and well trained operators possessing good knowledge about the plant [5]. The operator is trained on the simulated transients, malfunctions and abnormal conditions as in the real plant. This enables early detection of abnormalities, aiding in proper decision making and taking swift responses during crisis situation in the real plant.

In any NPP, efficient monitoring of the current status of the plant plays a key role in maintaining the equilibrium of the plant at each instance. Any imbalance in the plant is monitored and highlighted in the main control room (MCR) so that necessary steps are taken in time to avoid any fatal accidents. In such case, proper identification or classification of the occurrence of any event should be quick and unambiguous. To support such scenarios, transient identification systems have been devised to help operator identify transients and take fast and right corrective actions in due time [6]. Data driven methods such as artificial neural network and other soft computing techniques are used for transient identification in NPP [7].

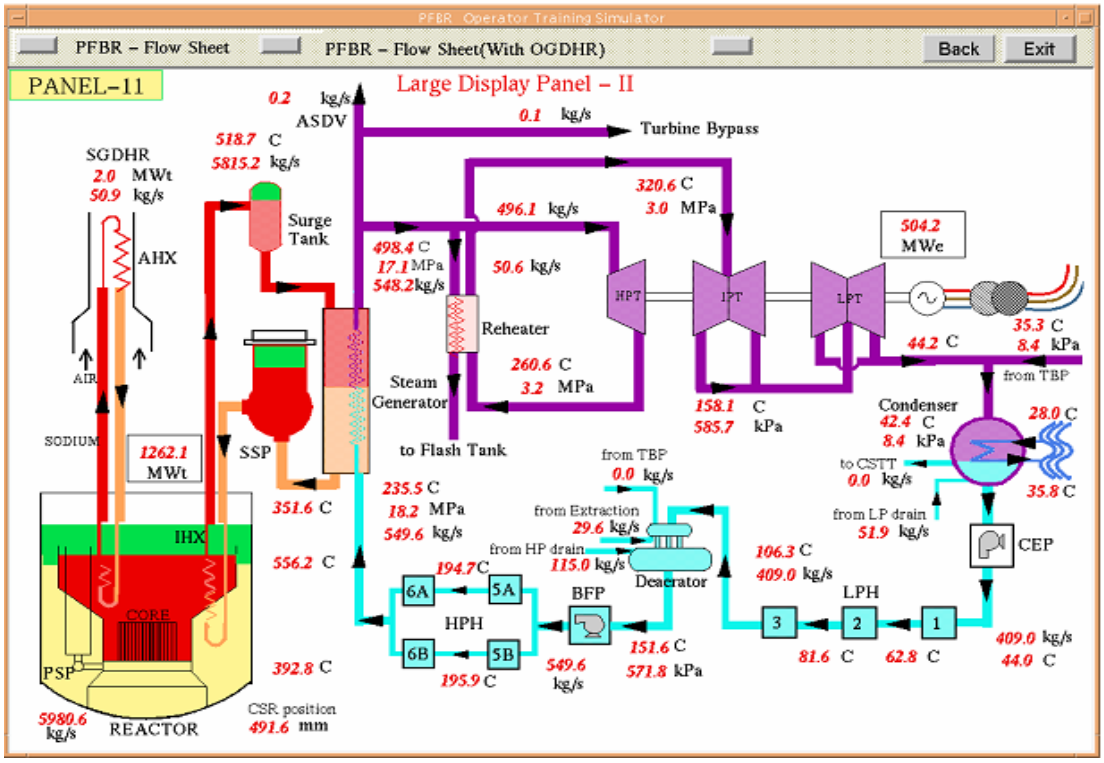


Figure 1.1 : Flow sheet of a 500MWe Fast Breeder Reactor

1.3 Event Classification

A NPP consists of a number of complex components which are integrated in a very systematic manner. Though the design of the plant has undertaken a rigorous planning and research, it is natural to have improper working of the components at some instance during the plant life span which may affect the balance of the plant. Such occurrence of conditions is known as designed basis events (DBE). These events may be a transient or a malfunction which may lead to vulnerable circumstances if not addressed on time. Prior to reacting to the event which has occurred, the classification of the correct event decides the course of action. The action to a particular event which actually never happened may lead to unnecessary chaos and eventually affect the balance and productivity of the plant. Hence, accurate event classification takes care of the safety along with economy and eventually the profitability factor of the plant.

Along with deterministic models which help in event identification and classification in NPP, various probabilistic models do assist in this regard. Some of the probabilistic approaches used are fuzzy computing, machine learning, and evolutionary computing. Firstly, fuzzy computing relates fuzzy logic to probabilistic reasoning in order to approach classification problems such as image processing [8], wireless sensor networks [9], data security [10], fault classification in transmission lines [11], etc. Fuzzy logic based modulation classification for non ideal environment has also been established where it is difficult to use probabilistic methods [12]. The usage of fuzzy rule based classification system (FRBCS) which is a category of fuzzy rule based system (FRBS) used for classification problems have been also extensively used and modified for usage in many diverse fields. Enhancing the performance of FRBCSs by extending the knowledge base with the application of the concept of Interval-Valued Fuzzy Sets is one such modification [13]. A learning algorithm based on reward and punishment has also been proposed to adjust the weights of each fuzzy rule in the rule-base by Jahromi et al. [14]. Pair wise learning and preference relations based linguistic FRBCS for solving multiclass problems are dealt by Fernandez et al. [15]. The study on the application of instance selection technique in genetic FRBCS is outlined by Fazzolari et al [16]. Ishibuchi et al. [17] reported the effect of rule weights on FRBCS. Secondly, machine learning is a mode of making computers behave as a human by providing a set of information which helps to find the possible outcome to a preferred input. Among the various diverse applications of machine learning, some of the most promising domains have been text categorization [18], medical diagnosis [19], data mining and information retrieval [20], etc. Lastly, evolutionary computing uses different principles of biological evolution such as natural selection and genetic inheritance to solve various problems including event classification. Some of the evolutionary computing techniques are genetic algorithm [21],

multi objective evolutionary algorithm [22], hybrid algorithm of linguistic classification rules and multi objective genetic algorithm [23], artificial bee colony [24], swarm intelligence [25], etc.

The fault detection and diagnosis methods are given much importance as they improve the safety, reliability and availability of NPP [26]. Event classification is one of the application areas of these methods. There have been a wide acceptance on the usage of probabilistic techniques along with many soft computing techniques to diagnose the event classification in NPP [27]. The feasibility study on transient identification using support vector machine (SVM) have been reported and it indicated that SVM classifiers showed promising results [28, 29]. Artificial neural network (ANN) has also showed its competence in various event identification [30] and transient classification [31]. A novel technique based on neural networks, aimed at reducing the variability of fault manifestations through a process of "intelligent normalisation" of transients for transient classification is reported by Roverso [32]. After a number of case studies, the use of back propagation algorithm for the development of connectionist expert system for transient identification in nuclear power plant is reported by Cheon et al. [33]. ANN has also been used for developing diagnostic systems for identification of various accident scenarios in NPP [34]. The classification of a transient as a “don’t know” transient if a classifier system do not have the accumulated knowledge regarding it, is certainly a wise way of reporting than incorrect classification [35]. This approach uses Kohonen’s self organizing map along with learning vector quantization instead of multilayer perceptron. The application of fuzzy logic based method for transient identification in NPP is reported by Marseguerra et al. [36]. Various adaptation to fuzzy logic based method has also been used to classify transient using optimized fuzzy clustering [37] and evolutionary fuzzy clustering [38]. One among all the innovative

approaches for transient classification is the ALADDIN methodology [39]. The ALADDIN approach combines three techniques for dynamic event and fault diagnosis in an attempt to improve the practical applicability and scalability of this type of system to real processes and machinery.

1.4 Problem Statement

The problem of classification of the occurrence of an event correctly and efficiently in a nuclear reactor holds the utmost priority. This classification system must produce results regarding occurrence of the events in a quick, correct and unambiguous manner. There is always a scope of improvement in achieving classification accuracy better than the previous best one. Moreover, nuclear reactor being safety critical systems, study on improving the classification accuracy becomes further more important. The present thesis delineates the usage of machine learning algorithms and soft computing techniques for such event classification in nuclear reactors. The generation of a massive volume of dataset in nuclear reactors increases the computational complexity of the classifiers. Though event classification in nuclear reactor have been studied using various techniques by many researchers, preprocessing of the dataset to reduce the computational cost have hardly been reported. The present work emphasizes on the usage of preprocessing of the dataset in an efficient manner prior to feeding it to a classifier so that it eventually improves the performance of the classifier. The preprocessing of the dataset results in reduction of the dimensionality and sample size of the dataset.

1.5 Objective and Scope of work

The present work demarcates some of the machine learning and soft computing algorithms for event classification in nuclear reactors. The events include some of the transient

and malfunctions occurring in the steam water system of a nuclear reactor. In a broader sense, this thesis aims at improving the NPP event classification system. The various objectives that have been set for the present research work are as follows:

- To check the performance of usage of soft computing techniques such as fuzzy logic for transient identification in nuclear reactors.
- To analyze the importance of feature selection for achieving better performance from the fuzzy logic systems.
- Study on the performance of supervised machine learning algorithms such as artificial neural network and adaptive neuro fuzzy inference system for multiclass transient classification in nuclear reactors.
- To develop an algorithm for preprocessing the dataset that reduces the training data samples and invigilating the performance of the classifier due to implementation of such approach.
- To study the feasibility of dimensionality reduction technique using principal component analysis for event identification in nuclear reactors.
- To address the issue of imbalanced dataset where the samples are not evenly distributed for each category of class that needs to be classified.
- To develop intelligent GUI based on machine learning algorithms and soft computing techniques.

The present research broadly aims at aiding a quick and unambiguous decision making process of the operator stationed at the main control room of nuclear reactors. It also helps in reducing the information overloading on the human operator due to the volume of information

being projected at him at every instance. The scope of the present work is diverse and global in nature as it can address similar issues from different domains of application.

1.6 Thesis Structure

The thesis is divided into a total of seven chapters. **Chapter 1** includes a brief introduction to NPP along with a detailed explanation on the need of event classification in NPP. Machine learning and its application in event classification problems are also elaborately described in this chapter. The problem statement which mostly addresses the event classification problem in nuclear reactors and its presentation using intelligent GUI are also explained clearly. The objective and scope of the research are discussed towards the end of this chapter.

Chapter 2 outlines two of the very popular techniques to solve event classification problem, i.e., ANN and FIS. This chapter also explains about some supervised machine learning algorithms for classification such as kNN and SVM. A collection of information regarding the various research carried out related to classification problems is also depicted. Towards the end of this chapter, the performance metrics of these classifiers are also elucidated.

Chapter 3 focuses on the concept of fuzzy logic used in the FIS in order to develop a FRBS for event identification. Two popularly used FRBS methods i.e. Mamdani type FRBS and TSK-type FRBS are explained in detail. The significance of generation of proper input and output membership functions along with an appropriate rule base and input features are delineated. A specific category of FRBS which is used for classification related problems is known as FRBCS. This chapter elucidates the possibility of usage of a simple FRBCS with optimized features for online event classification for a system which has very short cycle time.

Some of the transients from the steam water system (SWS) are considered as the events for experimentation.

Chapter 4 illuminates on a novel approach to reduce the number of training samples in a dataset named as training dataset reduction (TDR). The algorithm which governs this technique involves the creation of a hypothetical boundary based on a cut-off Euclidean distance (ED). The two methods adopted to find this cut-off ED named as mean-α-standard deviation method and area selection method are explained elaborately. A comparison is made on both these methods on some real world datasets and the result from these experiments is used to classify some of the transients in the SWS. A detailed inference on the practicability of usage of TDR for transient classification in nuclear reactors is summarized.

Chapter 5 explains the importance of dimensionality reduction of a dataset. One such dimensionality reduction algorithm named principal component analysis (PCA) is elucidated. The classifier used to classify the events is the adaptive neuro-fuzzy inference system (ANFIS) which has the advantage of both ANN and FIS. An elaborate description on this algorithm is presented along with its advantages and disadvantages. A feasibility confirmation on the performance of ANFIS classifier used for classifying some of the events in NPP considering PCA for dimensionality reduction is carried out and illustrated. A comparison on the usage of Multiple-ANFIS (MANFIS) with single ANFIS is also presented.

Chapter 6 deals with imbalanced dataset and the issues related to the usage of such datasets for classification purpose. Some of the most admired way of solving such problem is mentioned in this chapter, emphasizing on the oversampling technique. A detailed mechanism of operation of Synthetic minority oversampling technique (SMOTE), which is the popularly used

oversampling technique of addressing imbalanced dataset problem is expounded. Further, a modification to this approach named Weighted-SMOTE is illustrated. A comparison on the performance of the weighted-SMOTE and SMOTE is revealed using some real world datasets. The performance measures used to study such analysis are recall and F-measure as these two metrics exposes the real credibility of the classifier using an imbalanced dataset.

Chapter 7 projects the reason behind development of appropriate GUI, the various properties that should be taken care of during development of a GUI and the advantages of developing an intelligent GUI for an operator in various industries emphasizing on NPP. The developed GUI along with its various properties is elaborated in this chapter.

Chapter 8 summarizes the study and work carried out in the thesis and the conclusions drawn based on the research. This chapter ends by addressing the future scope of research.

2

BACKGROUND INFORMATION

The present chapter describes the background information on some of the supervised classification algorithms used in this thesis in detail. This chapter also elucidates on the various performance measures used to evaluate the performance of a classifier. This includes the confusion matrix, receiver operating characteristics and area under the receiver operating characteristics. An insight on the performance of a multi class classifier is also included in this thesis.

2.1 Fuzzy logic and fuzzy inference system

Fuzzy logic depicts the real world scenario in a more sensible approach using linguistic variables, overlapping classes and approximate reasoning. The need of fuzzy logic came into existence when people realized that everything in the world cannot be categorized or expressed in terms of only ‘Yes’ or ‘No’, ‘True’ or ‘False’ and ‘1’ or ‘0’. Zadeh [40] introduced the world about fuzzy logic and the concept of fuzzy set. This is a breed of revolution of its kind. Fuzzy logic uses variables which practically denote values which are not integers and produces an output with respect to the corresponding input. A membership function explains the fuzzy variables in a graphical manner. Fuzzy logic has a wide range of applications, including process controllers and event identification. Fuzzy logic is conceptually easy to understand as it uses simple english words in order to explain a particular set or domain.

A fuzzy inference system (FIS) or fuzzy rule base system (FRBS) is a data driven system in which a relationship is being established between the input and the output, based on a set of IF-THEN rules or otherwise called as fuzzy rules. The set of desired input-output numerical data pairs are represented as:

$$(x_1^{(1)}, x_2^{(1)}; y^{(1)}), (x_1^{(2)}, x_2^{(2)}; y^{(2)}), \ldots\ldots\ldots\ldots, (x_1^{(n)}, x_2^{(n)}; y^{(n)}) \quad \textbf{(2.1)}$$

where x_1 and x_2 are inputs and y is the output of a two inputs and one output system. The fuzzy rules are generated from these input output pairs which determines a mapping $f : (x_1 , x_2) \rightarrow y$ [41]. An example of a fuzzy rule in this kind of system is of the form

$$R^l : IF\ x_1^{(1)}\ is\ F_1^l\ and\ x_2^{(1)}\ is\ F_2^l\ THEN\ y^{(1)}\ is\ G^l \quad \textbf{(2.2)}$$

Here F_k^l represent the linguistic variable and $F_k^l \subset f(X_k)$, $k = 1, 2, 3, \ldots, n$, are the antecedent membership functions, and $G^l \subset f(Y)$ is the consequent membership function [42]. The input linguistic variables are denoted by x_k , $k = 1, 2, 3, \ldots n$ and the output linguistic variable is denoted by y. For classification using FRBS, the consequent part of the rule can be categorized into three varieties [43]. In the present thesis, the fuzzy rules with the class in consequent have been used for classification purpose. This approach is chosen because the FRBS would take data from a safety critical system which needs faster and interpretable outcomes. A general flow sheet depicting the different phases in the FIS is pictorially represented in Fig. Figure 2.1.

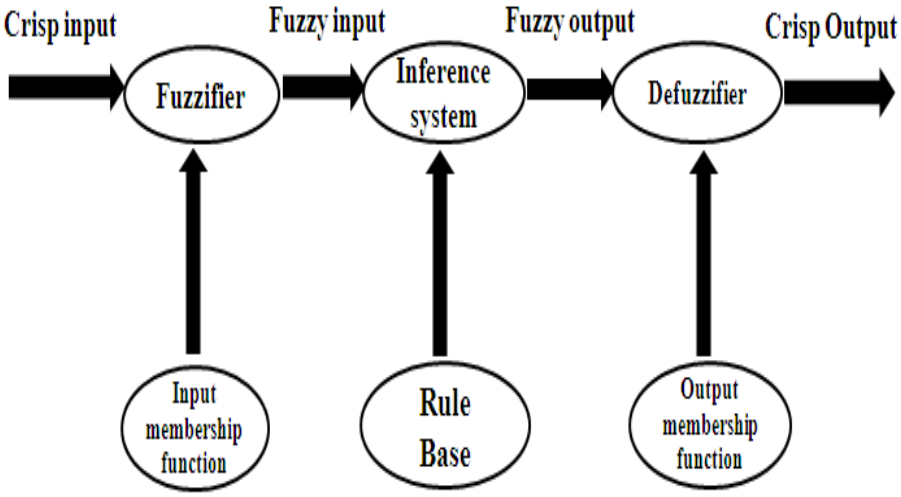


Figure 2.1: Flow sheet of a fuzzy inference system

2.1.1 Membership function (MF)

Universe of Discourse is the set that contains all the sets of interest for the given context problem. The membership function ($\mu_A(x)$) of the fuzzy set (A) maps the universe of discourse (X) on to the numerical values in the range [*0, 1*].

$$\mu_A(x) : X \rightarrow [0,1] \quad \textbf{(2.3)}$$

$$A = \{(x, \mu_A(x)) ; x \in X , \mu_A(x) \in [0,1]\} \quad \textbf{(2.4)}$$

2.1.2 Mamdani-type FRBS

There are mostly two types of FRBS. One of them is the Mamdani-type FRBS as shown in Fig.Figure 2. The following steps are involved in Mamdani-type FRBS approach. These are

Step 1: Segregation of the universe of discourse into fuzzy regions

Step 2: Generation of the fuzzy rules and rule base

Step 3: Processing of input output numerical data pairs

Step 4: Defuzzification procedure

- ***Step 1: Segregation of the universe of discourse into fuzzy regions***

This is the initial step where the domain interval of each fuzzy region is decided based on the input numerical data. The membership function for each input domain and output domain is constructed based on this segregation. The shape of the membership functions depends on the system designer who decides it based on the problem statement and the uncertainty present in it. Generally, the shape of the membership functions is triangular, trapezoidal, sigmoidal or Gaussian. The shape being constant in a particular problem for all the membership functions, the area of these regions may differ. This is based on the numerical data and the experience of the expert. Each segregated fuzzy domain is denoted by simple comparative phrases such as taller, much taller, somewhat taller etc for a domain representing height. Figure 2.3 shows an example where the input domains intervals x_1 and x_2 are divided into three and five regions respectively and the output domain interval y is divided into five regions with triangular shaped membership functions with different areas.

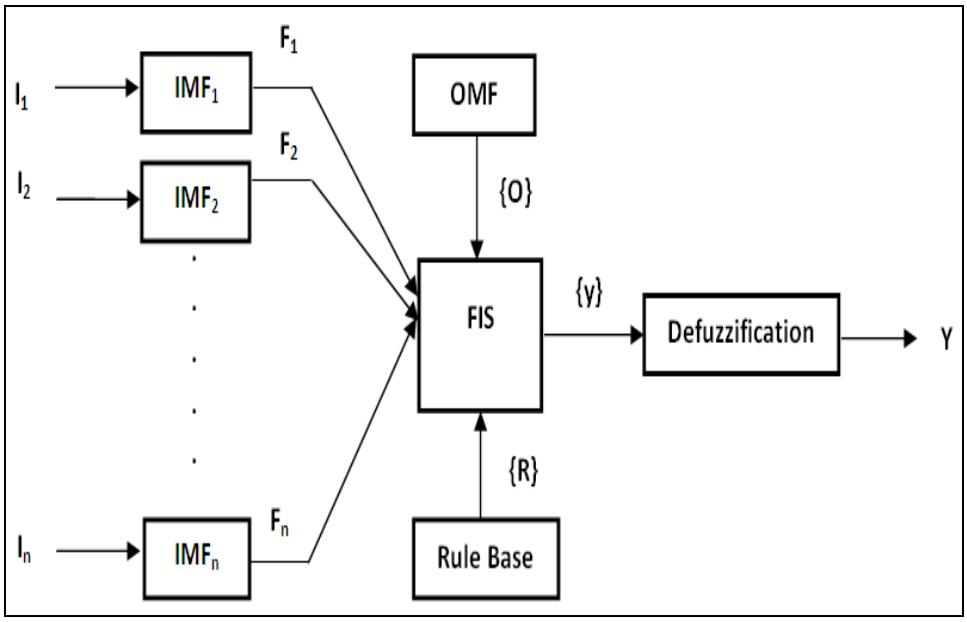


Figure 2.2: Block diagram of Mamdani-type fuzzy rule based system

Figure 2 explains the Mamdani-type fuzzy rule based system. Here, $I_1, I_2, \ldots, I_n$ are the set of numerical input data being mapped to their respective input membership functions $IMF_1, IMF_2, \ldots, IMF_n$ and produces their respective fuzzy input data, $F_1, F_2, \ldots, F_n$. The FIS produces the fuzzy output data y by mapping all the fuzzy input data on to the output membership function (OMF) using the corresponding rules R from the rule base. Y is the defuzzified numerical value of y.

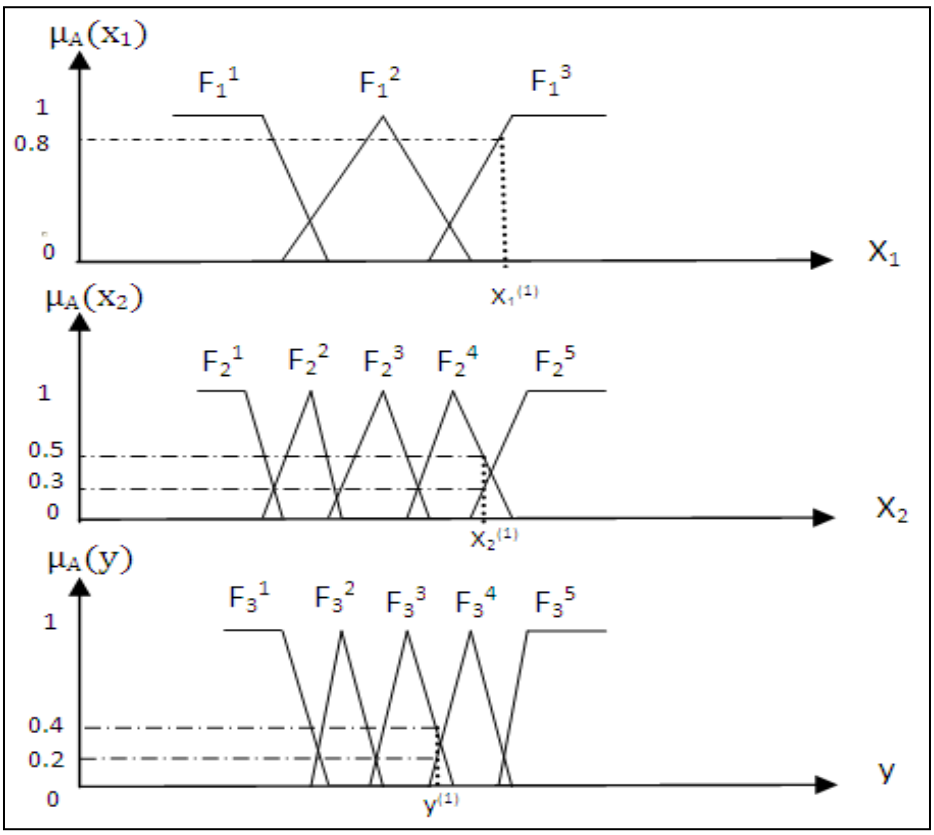


Figure 2.3: Diagrammatic representation of the membership functions operation in Mamdani type FRBS

- ***Step 2: Generation of the fuzzy rules and rule base***

The rules produce a relationship between the input domains and the output domain. This relationship is based on simple 'IF-THEN' statement. The 'IF-THEN' statement consists of an 'IF' part explaining the antecedent proposition and the 'THEN' part explaining the consequent

proposition. Both these parts can have multiple conditions separated by operators such as AND or OR. For a particular problem statement, each rule is based on a proper analysis on the input numerical data and the expert's experience. A rule based on these conditions as shown in Fig.Figure 2.3, could be of the form

$$R^1 : IF\ x_1\ is\ F_1^3\ and\ x_2\ is\ F_2^4\ THEN\ y\ is\ F_3^3 \quad (2.5)$$

The rule R^1 generated here is based on the problem statement which demanded an AND operation or otherwise it could have been an OR also. This depends on the context of the problem statement. All the rules are generated likewise based on the numerical data and the fuzzy regions with the expert's experience taken into consideration.

The representation of these rules is done in a matrix form known as fuzzy rule base. This representation becomes difficult when the number of input parameters is more. The choice of the number of parameters and the rule base decide the interpretability and accuracy of the FRBS. Interpretability of fuzzy systems is an ability to explain the behavior of the system in an understandable way [44]. Increase in the number of input parameters need not always lead to a better accuracy of the output. This is because fuzzy models and controllers cannot usually have a large number of variables without falling prey to the curse of dimensionality [44]. Another way of approaching it is to make the rules more concise and subtle covering all the aspects of the particular problem statement. The best option for a FRBS would be to have a rule base covering the entire input and output domain efficiently using minimum number of input parameters and the output being within the output error margin. This should satisfy the tradeoff between the interpretability and the accuracy of the FRBS.

- ***Step 3: Processing of input output numerical data pairs***

After the generation of all the membership functions and the rule base, the input numerical data is mapped with the output numerical data based on the membership functions and rule base. The input numerical data is initially mapped on to the respective membership functions and their specific membership degree is found. As described earlier in Step 2, a degree is assigned to each input numerical data based on their respective membership function. Now, based on rule governing that particular input numerical data, the resultant of the mapped degree calculated from the antecedent proposition is mapped on to the output membership function. This is concurrently done for all the input membership functions and finally a number of clipped output membership functions are collected. Figure 2.3 shows that after mapping $x_1^{(1)}$ on F_1 series of membership function, the first stage output degree is found to be ($0F_1^1$, $0F_1^2$, $0.8F_1^3$). Similarly for mapping $x_2^{(1)}$ on F_2 series of membership function, the second stage output degree is found to be ($0F_2^1$, $0F_2^2$, $0F_2^3$, $0.5F_2^4$, $0.3F_2^5$). The rule R^1 has AND operator in the antecedent proposition. Hence, the minimum degree from all the combinations of the mapped input membership function is considered. The combination which has no consequent proposition is considered as zero or the rule does not exist. The final degree for each case is now mapped on to the output membership function based on the rule base. Finally, a bunch of clipped output membership functions are collected and aggregated.

- ***Step 4: Defuzzification procedure***

The aggregated output membership functions go through a stage called as defuzzification where these collective information needs to be converted to a numerical data for usage in real systems. There are many defuzzification processes such as Centroid method, bisector method,

middle of maximum, smallest of maximum, largest of maximum, etc. The easiest and arguably the best one is the Centroid method. The final output (Y) from this method is calculated as

$$\boldsymbol{Y = \frac{\sum_{i=1}^{K} y^i \, x \, \mu_{A^i}^i(y_i)}{\sum_{i=1}^{K} \mu_{A^i}^i(y_i)}} \quad \textbf{(2.6)}$$

Here y^i denotes the center value of A^i (the center of a fuzzy region is defined as the point that has the smallest absolute value among all the points at which the membership function for this region has membership value equal to one) and K is the number of fuzzy rules in the fuzzy rule base [44].

2.1.3 Takagi-Sugeno-Kang (TSK) type FRBS

Takagi-Sugeno-Kang (TSK) type FRBS is almost similar to Mamdani-type FRBS with some changes. One of the changes is done at the generation of the rules. The antecedent proposition of the rule is composed of linguistic variables but the consequent proposition is represented as a function of the input variables [45]. For an input-output pair ($x_1^{(1)}$, $x_2^{(1)}$; $y^{(1)}$), say the rule is as follows

$$\boldsymbol{R^1 : IF\ x_1^{(1)}\ is\ F_1^1\ and\ x_2^{(1)}\ is\ F_2^1\ THEN\ y^{(1)} = f_1(x_1^{(1)}, x_2^{(1)})} \quad \textbf{(2.7)}$$

$$\boldsymbol{R^2 : IF\ x_1^{(1)}\ is\ F_1^2\ and\ x_2^{(1)}\ is\ F_2^2\ THEN\ y^{(2)} = f_2(x_1^{(1)}, x_2^{(1)})} \quad \textbf{(2.8)}$$

Here, $f_1(x_1^{(1)}, x_2^{(1)}) = p_1 x_1^{(1)} + q_1 x_2^{(1)} + r_1$ and $f_2(x_1^{(1)}, x_2^{(1)}) = p_2 x_1^{(1)} + q_2 x_2^{(1)} + r_2$ where (p_i, q_i, r_i) is a vector of real numbers. The resulting output (Y) is based on the degree of applicability, i.e., λ_1 and λ_2 for the inputs, i.e., $x_1^{(1)}$ and $x_2^{(1)}$ respectively. The resulting output is

$$\boldsymbol{Y = \frac{\lambda_1 y^{(1)} + \lambda_2 y^{(2)}}{\lambda_1 + \lambda_2}} \quad \textbf{(2.9)}$$

In general representation of the resulting output with *R* number of rules is as follows

$$\boldsymbol{Y = \frac{\sum_{i=1}^{R} \lambda_i y^{(i)}}{\sum_{i=1}^{R} \lambda_i}} \qquad \textbf{(2.10)}$$

Here, the defuzzification stage is bypassed with a weighted average in order to get a crisp output. So this does not have an output membership function. The expressive power and interpretability of Mamdani output is lost in the Sugeno FIS since the consequents of the rules are not fuzzy [46]. For this reason, Mamdani-type FRBS is used widely as compared to TSK-type FRBS. But TSK-type FRBS has a better processing time as the weighted average replaces the time consuming defuzzification process [47].

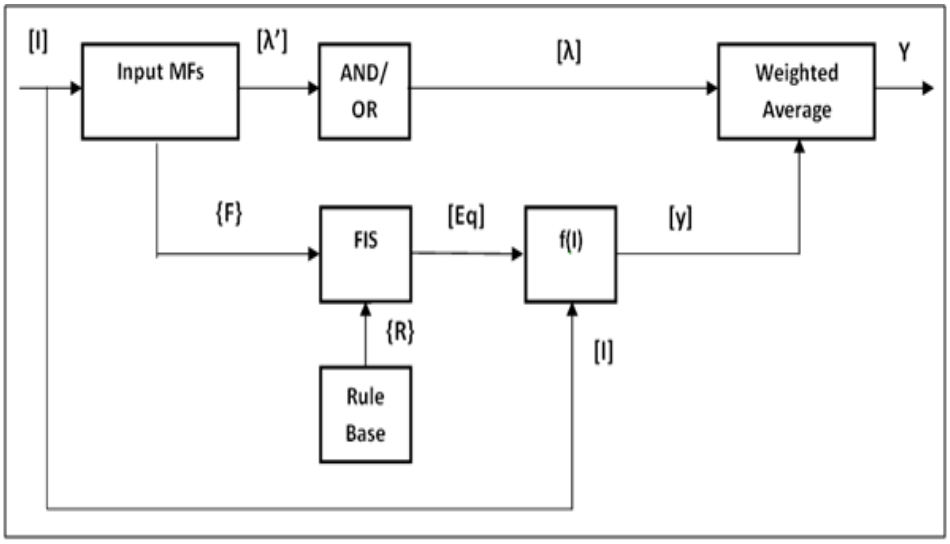


Figure 2.4 : Block diagram of Takagi-Sugeno-Kang-type fuzzy rule based system

Figure 2.4 shows the diagrammatic representation of the TSK-type FRBS. Here, *I* represents the set of numerical input variables each cycle and *F* being the fuzzified value of those *I*. *λ'* denotes the degree of applicability and *λ* denotes the final degree of applicability. The fuzzy input data *F* and the selected rule from the rule base *R* are fed to FIS. *Eq* represents the set of

equations selected based on rule *R* and fuzzy input data *F*. The substitution of numerical input data *I* on to the equations *Eq* is done by *f(I)* which produces the numerical output data *y*. The final degree of applicability λ along with numerical output data *y* generates the final output data. This is the approach undertaken by TSK-type FRBS for classification problems.

2.2 Supervised Machine Learning Algorithms

Supervised machine learning (SML) is a machine learning methodology where a training set of data with known input and output labels, is used to train a system for unknown or test data. The output label of the test data is found by mapping a test data on to a function inferred from the training data. Some of the very common SML algorithms used for classification problems are k-nearest neighbor (kNN), support vector machine (SVM), artificial neural networks (ANN), etc. Unsupervised machine learning and reinforcement learning are the two other broad categories of machine learning. In unsupervised machine learning, labels are not available; hence the feature learning is done on its own by finding clusters in the inputs. Unsupervised learning algorithms according to Ghahramani [48] are used to find structures from data samples. Some of the examples of the unsupervised learning algorithms are self-organizing map (SOM), adaptive resonance theory (ART), k-means clustering, etc. The reinforcement learning is a process in which the system learns by interacting with the environment and learning gets modified based on the feedback from the interaction with the environment. According to Kaelbling et al. [49] , reinforcement learning is the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment. Some of the reinforcement algorithms are temporal difference learning, Q-learning, state-action-reward-state-action (SARSA), etc. This thesis focuses only on SML and some of the algorithms based on this concept.

2.2.1 k-nearest neighbor

The kNN algorithm is a supervised learning algorithm which classifies a query or a test data based on the k-nearest training data taken as reference as presented in Fig.Figure 2.5. It basically has two phases: Training phase and Classification phase. In training phase, a previously collected training data containing a multi dimensional input attributes along with their respective output labels is taken as reference and mapped. In the classification phase, the test data is mapped based on its input attributes and the output label is determined by taking a vote from the k-nearest neighbors of that test data. The nearest neighbor is found out by finding the distance between the test data and each of the training data individually and finding the k-least distance training data. The accuracy of the kNN approach heavily depends on the metric used to compute the distance between two samples [50]. The usage of Mahanabolis distance metric for kNN classification by semi definite programming is shown by Weinberger et al. [51]. This algorithm proves to be very effective, in terms of reducing the misclassification error, when the number of samples in training dataset is large [52]. Another advantage of the kNN method over many other supervised learning methods like SVM, decision tree, ANN, etc., is that it can easily deal with problems in which the class size is three and higher [53].

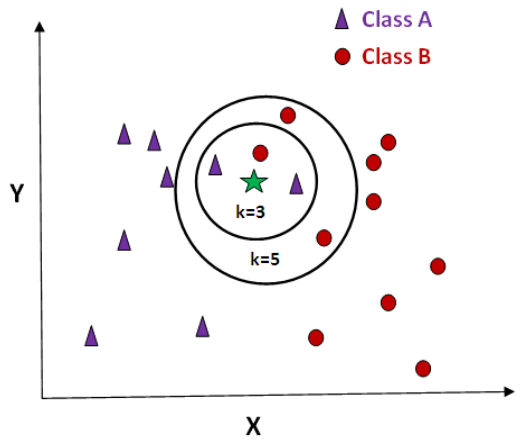


Figure 2.5: Classification using kNN Algorithm

Consider a dataset X with consisting R number of rows and C number of columns. Each row is a sample denoted as $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iC}\}$. Hence the resultant column matrix of the dataset $X = [X_1, X_2, \ldots, X_R]^T$. Similar is the case with another dataset Y = $[Y_1, Y_2, \ldots, Y_R]^T$ where $Y_j = \{y_{j1}, y_{j2}, \ldots, y_{jC}\}$. The distance between the vectors X_i and Y_j is calculated in different manners out of which some are described below.

- City block metric

$$\boldsymbol{D_{ij} = \sum_{k=1}^{C} \left| X_{ik} - Y_{jk} \right|} \quad \textbf{(2.11)}$$

- Euclidean distance

$$\boldsymbol{D_{ij}^{\,2} = (X_i - Y_j)(X_i - Y_j)^T} \quad \textbf{(2.12)}$$

- Minkowski metric

$$\boldsymbol{D_{ij} = \sqrt[p]{\sum_{k=1}^{C} \left| X_{ik} - Y_{jk} \right|^p}} \quad \textbf{(2.13)}$$

This shows that for p=1; Minkowski metric gives City block metric.

It also shows that for p=2; Minkowski metric gives Euclidean distance.

- Mahalanobis distance

$$\boldsymbol{D_{ij}^{\,2} = (X_i - Y_j)V^{-1}(X_i - Y_j)^T} \quad \textbf{(2.14)}$$

V is the covariance matrix

2.2.2 Support vector machine

SVM [54] is a supervised learning algorithm which is mostly used for classification or regression analysis**.** A SVM constructs an optimal hyper plane (OHP) with the largest distance

between the nearest training data of opposite class called as support vectors as shown in Fig. Figure 2.6. This distance is called functional margin which is inversely proportional to the generalization error of the classifier [55]. According to Roobaert [56], let the training data is given as $\{(x_1,y_1), (x_2,y_2), \ldots.., (x_l,y_l)\}$, where $x_i \in R$ and $y_i \in \{1, -1\}$, then the OHP, $w \bullet x + b = 0$, can be found by minimizing

$$\frac{1}{2}\|\mathbf{w}\|^2$$

constrained by: $y_i(w_i \bullet x_i + b) \geq 1$ for i = 1, 2, ... , l

This optimization problem can be solved by finding the saddle point of the Lagrangian 'L' using the method of Lagrange multipliers where

$$\boldsymbol{L = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i y_i (w_i \bullet x_i + b) + \sum_{i=1}^{l} \alpha_i} \quad \textbf{(2.15)}$$

Here, α_i the non-negative Lagrange multiplier. This is equivalent to find the saddle point in the dual formation by maximizing L_{dual}

$$\boldsymbol{L_{dual} = -\frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i \bullet x_j) + \sum_{i=1}^{l} \alpha_i} \quad \textbf{(2.16)}$$

with constraints : $0 \leq \alpha_i$ for i = 1, 2, ... , l

and $\sum_{i=1}^{l} \alpha_i y_i = 0$

This is a quadratic programming problem and the solution of this problem, w*, can be written as a linear combination: $w^* = \sum_{i=1}^{l} \alpha_i y_i x_i$

Another way of getting the maximum margin hyperplane is by creating non linear classifiers using kernel trick. Some commonly used kernels are polynomial kernel, Gaussian radial basis function kernel and hyperbolic tangent kernel whose mathematical equations are

mentioned below where *d, γ, α* and *β* are parameters of the kernel. Multi kernels SVM [57] are also used with high accuracy and great generalization.

- Polynomial kernel

$$\boldsymbol{K(X,Y)} = (\boldsymbol{X} \cdot \boldsymbol{Y} + \mathbf{1})^{\boldsymbol{d}} \tag{2.17}$$

- Gaussian radial basis kernel

$$\mathbf{K(X,Y)} = \mathbf{exp}(-\boldsymbol{\gamma}\|\mathbf{X} - \mathbf{Y}\|^{\mathbf{2}}) \tag{2.18}$$

- Hyperbolic tangent kernel

$$\boldsymbol{K(X,Y)} = \boldsymbol{tanh}(\boldsymbol{\alpha}(\boldsymbol{X} \cdot \boldsymbol{Y}) + \boldsymbol{\beta}) \tag{2.19}$$

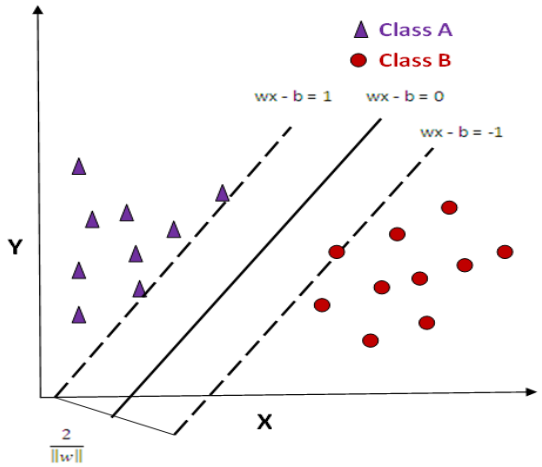


Figure 2.6: Basic decision boundary learned by support vector machine

2.2.3 Artificial neural network

ANN is a family of computational models based on biological nervous system used for information processing. Here, information processing mostly emphasizes on estimating or approximating functions that are dependent on input data. Figure 2.7 illustrates the structure of a basic ANN model. The model has a structured layer with the starting layer being the input layer and the end layer being the output layer. Both these layers are interconnected by one or more

hidden layers. Each layer consists of a number of nodes which are otherwise known as artificial neuron. An artificial neuron resembles in a similar fashion as that of the biological neuron. A biological neuron receives signal through a specific region in the dendrites known as synapses. On receiving a signal higher than the threshold value, the neuron gets activated and it transmits a signal through its axon to another neuron which might activate it.

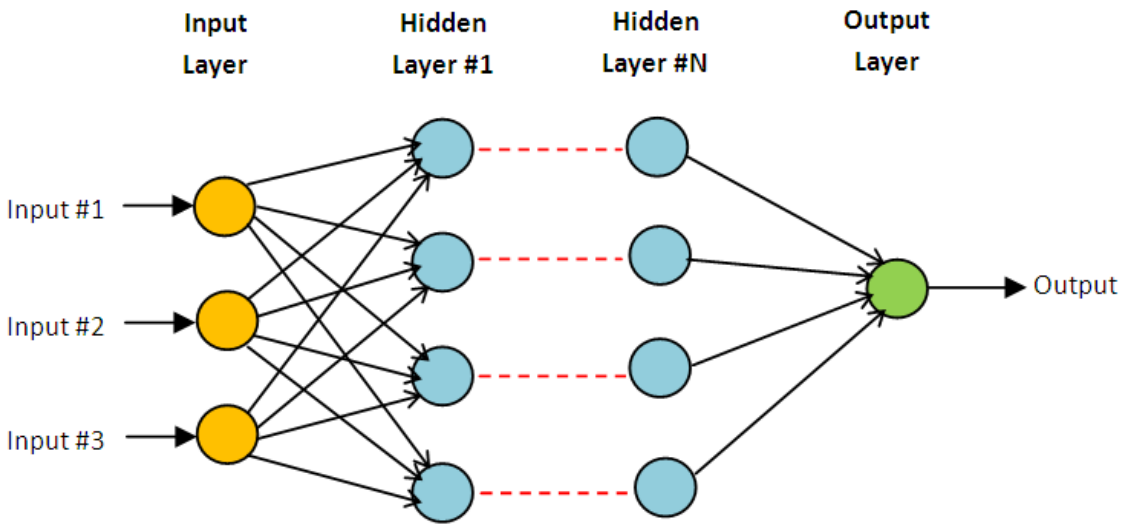


Figure 2.7: Architecture of a basic artificial neural network

Figure 2.8 illustrates the structure of an artificial neuron as modeled by McCulloch and Pitts [58].

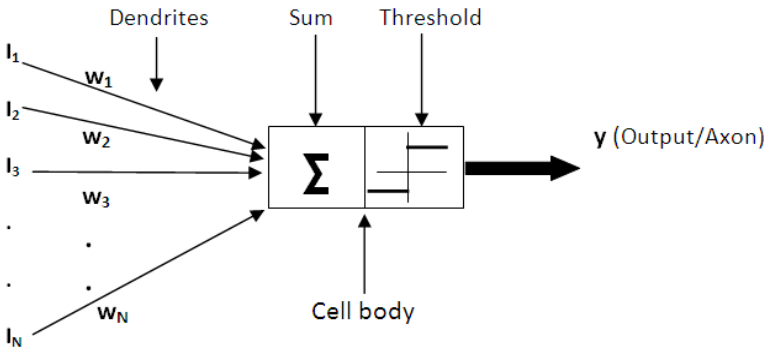


Figure 2.8: A basic artificial neuron model

A basic artificial neuron basically consists of two sections; sum and threshold. The sum section adds the input values with their respective weights. The threshold section consists of the activation function or otherwise known as the transfer function which takes the triggering decision of the neuron. There are broadly two categories of activation functions, i.e., linear activation function and non-linear activation functions. Non linear activation functions keep the response of the neuron in the bounded region. Figure 2.9 describes some of the commonly used activation functions like linear, step, Gaussian, sigmoidal and tan-hyperbolic activation functions.

- Linear activation function

$$\boldsymbol{f(x) = x} \tag{2.20}$$

- Step activation function

$$\boldsymbol{f(x) = sgn\,(x)} \tag{2.21}$$

- Gaussian activation function

$$\boldsymbol{f(x) = exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)} \tag{2.22}$$

- Sigmoidal activation function

$$\boldsymbol{f(x) = \frac{1}{(1 + exp(-\beta x))}} \tag{2.23}$$

- Tangent-hyberbolic function

$$\boldsymbol{f(x) = tan\,h(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)}} \tag{2.24}$$

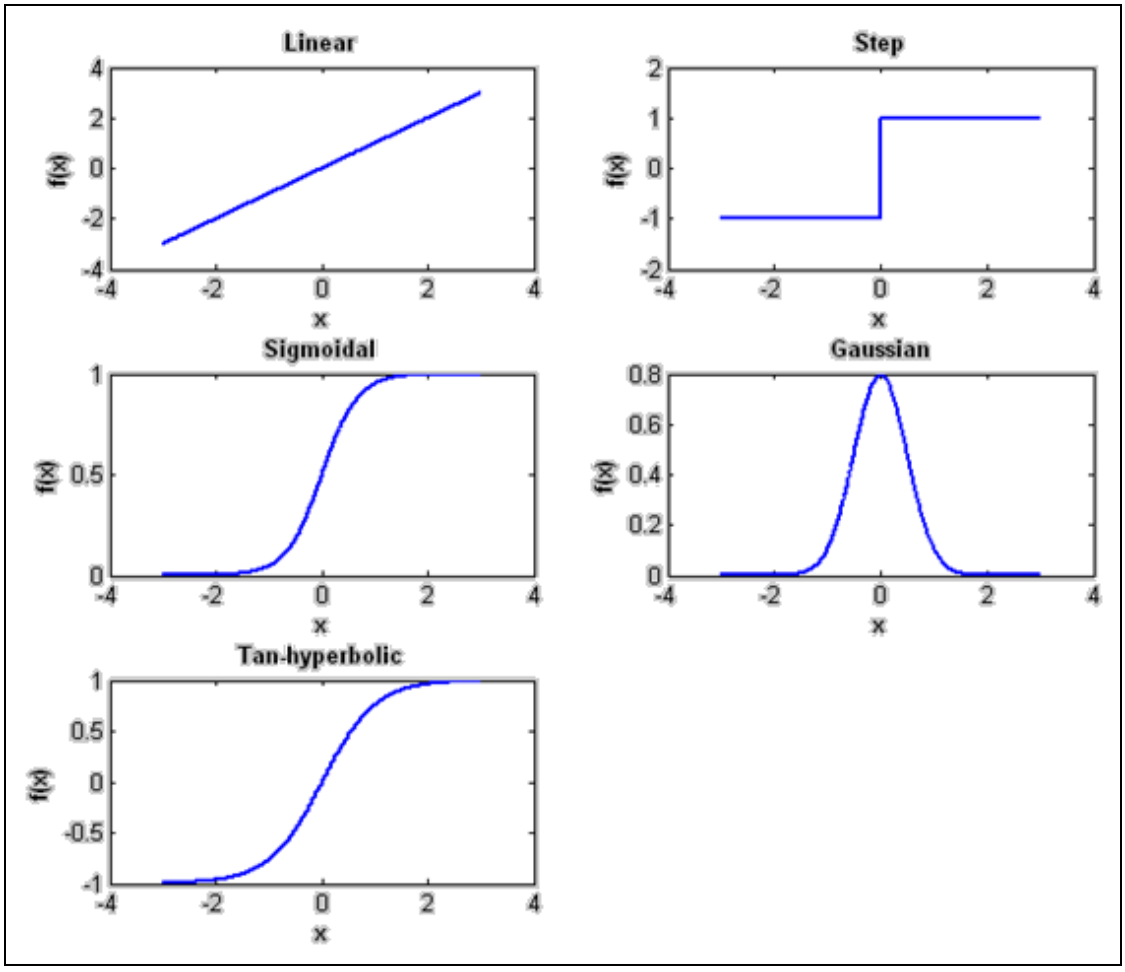


Figure 2.9: Commonly used activation functions

Neural networks are broadly classified into feed-forward networks and recurrent or feedback networks. The feed-forward neural networks do not form a loop or feedback among the different layers and the data passes through layer by layer and produces output at the output layer. The recurrent or the feedback network is a section of ANN where the network connections are feedback in nature. This approach allows the network to reveal dynamic temporal behavior which results in its application an unsegmented connected handwriting recognition or speech recognition. The feed-forward neural networks are further segregated as single layer perceptron, multilayer perceptron and radial basis function neural network. The recurrent neural network is classified into competitive networks, Kohonen's self organizing map, Hopefield network and

adaptive resonance theory models. This thesis emphasizes on feed-forward networks for the classification problems.

The operation of feed-forward network is divided into two categories, i.e., learning phase and testing phase. In the learning phase or otherwise called as training phase, a supervised learning algorithm is undertaken to train the classifier which is the feed-forward neural network over here. The input patterns are passed through each layer in a transformed manner using the weights assigned to each connection. Eventually, after crossing through these layers, the final output of an input pattern is produced and compared with the ideal output in the output layer. The units in the output layer belong to different categories. Ideally, the correct category output should have the largest value compared to all the other output values which should be comparatively very small. The comparison of the calculated output and actual output helps in modifying the weights assigned to each connection of neurons by a procedural algorithm. This modification in weights aims at decreasing the error between each output layer actual value and calculated value during each epoch. This procedure continues till an appropriate stopping criterion is valid. These stopping criteria might be achieving the necessary threshold error or reaching the maximum number of epochs that the system should undergo in this procedure. Once either of the stopping criteria is satisfied, the learning phase culminates and it is said that the feedforward network is trained. This gives the final weights of each connection of neurons which cannot be further modified. The testing phase commences after the learning phases culminates. Each of the test data which has not been used as the training dataset is fed as input to the trained neural network one at a time. The output of each test data is calculated and compared with the corresponding actual output. Once all the test data's outputs are calculated and

compared with their corresponding actual test output, the performance of the classifier is calculated.

Multilayer perceptron (MLP) is a feedforward ANN model that works on the principle of supervised machine learning algorithm to map a set of input data to its respective output. A MLP consists of one or more hidden layers along with the input and output layer which are fully connected to the preceding and the following layer as depicted in Fig.Figure 2.10. The hidden layers and the output layers consist of non-linear activation function. Backpropagation is the supervised machine learning algorithm which MLP uses for training process. Backpropagation algorithm propagates the error gradient or loss function gradient in a reverse direction during the training process which gets used in the next iteration. This process continues in each of the iterations till the error reaches the threshold error margin. The activation function during the backpropagation must be differentiable in order to proceed with this training process. Hence, generally a sigmoidal activation function is used in MLP. MLP is competent enough to learn non-linear models and online learning models. Due to the non convexity of the gradient of the loss function, the global maximum is not guaranteed using backpropagation with gradient descent method for optimization. This approach also needs optimization of the hyperparameters such as number of hidden neurons, layers and iterations. It is reported that multilayer feed forward networks with as few as one hidden layer are a class of universal approximators [59].

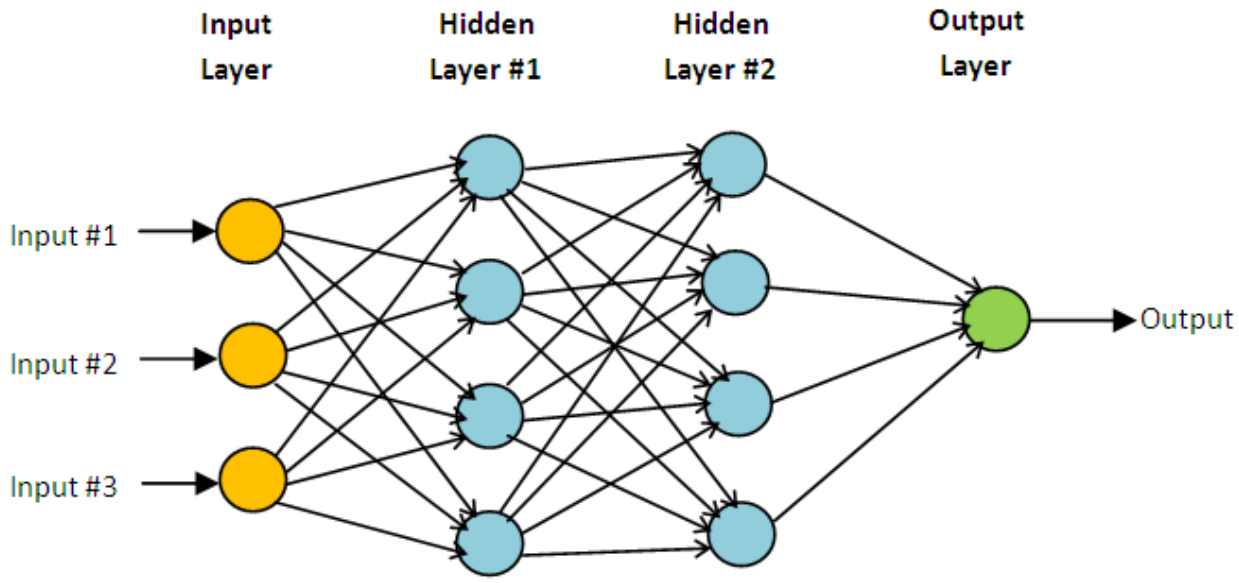


Figure 2.10: Architecture of a multilayer perceptron

Radial basis function (RBF) network is a feed forward ANN with only one hidden layer using radial basis function as the activation function. The output layer is a linear summation function. Figure 2.11 depicts the commonly used RBF using the Gaussian function. Initially, the training process of the RBF network undergoes calculating the spread of the input space about the centroid of the activation function. This is done by using an unsupervised learning algorithm, mostly k-means clustering. Later, the weights from the hidden to output layer are updated using least square function or singular value decomposition. The major two drawbacks of RBF networks are (i) its performance with noisy data is very poor and (ii) its computationally inefficient performance for larger dataset because of only one hidden layer. In order to improve the performance there has been a lot of modifications to RBF network. . RBF network is useful in function approximation, classification and modeling of dynamic systems and time series [60]. One among them is replacing the Gaussian activation function to sigmoidal activation function

for better capability of approximation, faster learning speed, better size of network and high robustness to outliers [61].

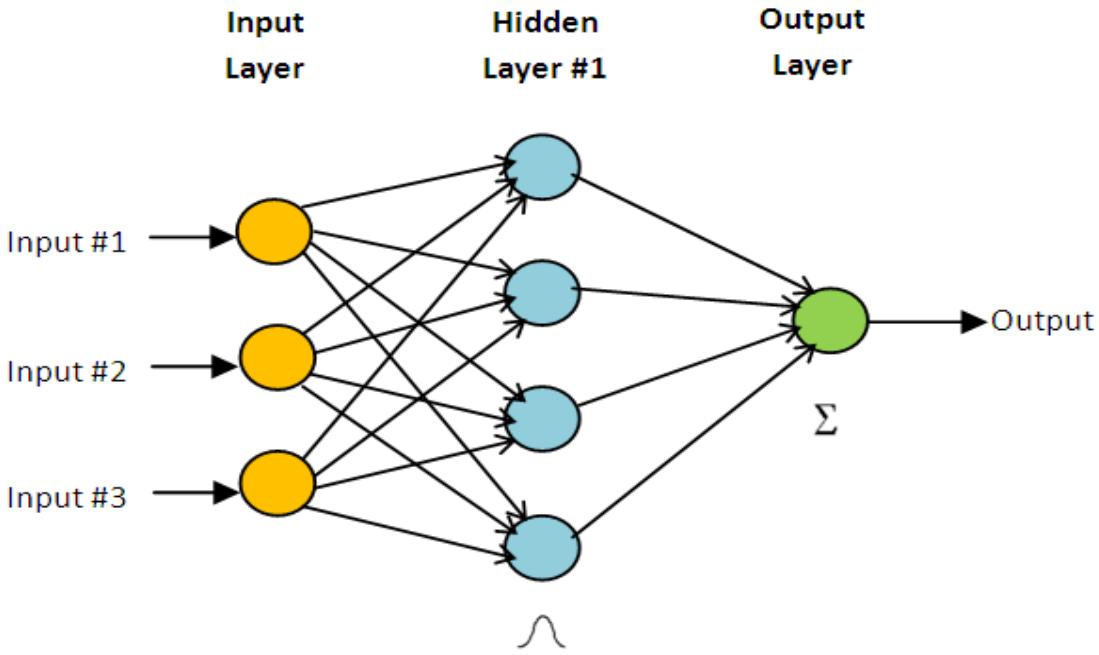


Figure 2.11: Architecture of a radial basis function network

As mentioned earlier, backpropagation algorithm is the common learning algorithm used for training the neural networks. The step wise approach of this algorithm is outlined below [62].

I. The weights of each node to node connection are initialized to small random values.

II. From the training dataset, randomly an input pattern $x^{(i)}$ is chosen.

III. The signal corresponding to this selected input pattern is propagated forward through all the layers (l = 1 to L) in the network.

IV. The error δ_i^l at the output layer is calculated considering the difference in the actual output d_i^l and the calculated output y_i^l using the net input h_i^l at the i^{th} unit of the l^{th} layer along with the derivative of the activation function.

$$\boldsymbol{\delta_i^L = g'(h_i^L)\left[d_i^L - y_i^L\right]} \qquad \text{here , } l = L \qquad \textbf{(2.25)}$$

V. This error is propagated backwards and the deltas at each nodes are calculated

$$\boldsymbol{\delta_i^l = g'(h_i^l)\sum_j w_{ij}^{l+1}\delta_i^{l+1}} \qquad \textbf{for } \boldsymbol{l} \textbf{ = } \boldsymbol{(L}\textbf{-1), . . ., 1} \qquad \textbf{(2.26)}$$

VI. The weights are also updated in each layer in a backward direction based on a particular learning rate η.

$$\boldsymbol{\Delta w_{ji}^l = \eta \delta_i^l y_j^{l-1}} \qquad \textbf{(2.27)}$$

$$\boldsymbol{w_{ji}^l = w_{ji}^l + \Delta w_{ji}^l} \qquad \textbf{(2.28)}$$

VII. The procedure from step II is repeated for the next randomly selected input pattern until any of the stopping criterion gets satisfied.

2.3 Performance metrics

A confusion matrix is an important measure to check the performance of a classifier network learned using any machine learning algorithm. Figure 2.12 shows the representation of a confusion matrix for a binary classification problem. True Negatives (TN) are the number of negative examples which are correctly classified as negative. False Positives (FP) are the number of negative examples which are incorrectly classified as positive. True Positives (TP) are the number of positive examples which are correctly classified as positive. False Negatives (FN) are the number of positive examples which are incorrectly classified as negative. In the present thesis, the majority class samples are the negative class samples and the minority class samples are the positive class samples.

	Predicted Negative	Predicted Positive
Target Negative	TN	FP
Target Positive	FN	TP

Figure 2.12: Basic **r**epresentation of a confusion matrix for a binary classifier

The most commonly used performance measures derived from the confusion matrix are accuracy and error rate. Accuracy is the ratio of the number of all the correctly classified samples to the total number of test samples. Error rate is the ratio of all the incorrectly classified samples to the total number of test samples. It is quite obvious that accuracy and error rate sum up to 1.

$$\boldsymbol{Accuracy} = (\boldsymbol{TN} + \boldsymbol{TP})/(\boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN} + \boldsymbol{TP}) \quad \textbf{(2.29)}$$

$$\boldsymbol{Error\ rate} = (\boldsymbol{FP} + \boldsymbol{FN})/(\boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN} + \boldsymbol{TP}) \quad \textbf{(2.30)}$$

Accuracy and error rate measures are very deceptive as these are data dependent. In case of imbalanced dataset where the number of majority samples is too high compared to the minority samples, the classifier gets biased towards the majority samples. In such cases, the accuracy metric results in a very high value and the error rate being very less. These results project as if the classifier is an ideal one which actually is not the real scenario. In such cases, the minority class does not get properly classified yet these two metrics suggest the classifier to be an efficient one. In order to overcome such imbalanced dataset scenarios, there are some other evaluation metrics which states the actual performance of the classifier. These metrics are

precision, recall, specificity, fall-out, F-measure and G-mean. These measures are calculated as follows.

$$\boldsymbol{Precision} = \boldsymbol{TP}/(\boldsymbol{TP} + \boldsymbol{FP}) \quad \textbf{(2.31)}$$

$$\boldsymbol{Recall\ or\ Sensitivity\ or\ True\ Positive\ Rate} = \boldsymbol{TP}/(\boldsymbol{TP} + \boldsymbol{FN}) \quad \textbf{(2.32)}$$

$$\boldsymbol{Specificity\ or\ True\ Negative\ Rate} = \boldsymbol{TN}/(\boldsymbol{TN} + \boldsymbol{FP}) \quad \textbf{(2.33)}$$

$$\boldsymbol{Fall\text{-}out\ or\ False\ Positive\ Rate\ or}\ (\mathbf{1} - \boldsymbol{Specificity}) = \boldsymbol{FP}/(\boldsymbol{TN} + \boldsymbol{FP}) \quad \textbf{(2.34)}$$

$$\boldsymbol{F\text{-}Measure} = \frac{((\mathbf{1} + \boldsymbol{\beta})^{\mathbf{2}} \times \boldsymbol{Recall} \times \boldsymbol{Precision}}{(\boldsymbol{\beta}^{\mathbf{2}} \times \boldsymbol{Recall}) + \boldsymbol{Precision}} \quad \textbf{(2.35)}$$

$$\boldsymbol{G\text{-}Mean} = \sqrt{\boldsymbol{Precision} \times \boldsymbol{Recall}} \quad \textbf{(2.36)}$$

In Eq. 2.35, β is the coefficient to adjust the relative importance of precision and recall. Usually precision and recall have equal importance so β = 1. F-measure is the harmonic mean and G-mean is the geometric mean of precision and recall. So, Eq. 2.35 can be simplified as follows.

$$\boldsymbol{F\text{-}Measure} = \frac{\mathbf{2} \times \boldsymbol{Recall} \times \boldsymbol{Precision}}{\boldsymbol{Recall} + \boldsymbol{Precision}} \quad \textbf{(2.37)}$$

Precision is the measure of the exactness of the samples which are correctly classified positive out of the samples which are classified positive. Recall is the measure of the samples which are correctly classified positive out of the samples which are actually positive. Recall is also known as sensitivity or true positive rate. The importance and information of both precision

and recall can be combined together as a measure known as F-measure. F-measure is the harmonic mean where as G-mean is the geometric mean of precision and recall. Specificity is the measure of the samples which are correctly classified negative out of the samples which are actually negative. This is similar to sensitivity. It is also called true negative rate. Fall-out is the measure of the samples which are incorrectly classified positive out of all the samples which are actually negative. Fall-out is also called false positive rate or 1-specificity. In case of imbalanced dataset where a particular class has the minority of the samples compared to others, precision, recall and F-measure defines the performance of the classifier. In those cases, only accuracy does not provide the correct conclusion about the performance of the classifier. More about such cases is explained in Chapter 5 of the present thesis which elucidates on the imbalanced dataset problem and its solutions.

Another way of analyzing the performance of a classifier is by plotting a receiver operating characteristics (ROC) curve which is a two-dimensional depiction of the classifier's performance [63]. ROC curves are used to judge the discrimination ability of various statistical methods that combine various clues, test results, etc for predictive purposes [64]. The ROC curve is the plot between the false positive rate (FPR) in the X-axis and true positive rate (TPR) in the Y-axis. TPR is also known as sensitivity and FPR is known as (1-specificity). Specificity is the measure of the percentage of the negatives that are properly classified as negatives. Visually, the more ROC shifts towards the left corner of the curve, i.e., towards the Y-axis, the better is the classifier. This is a very simple and efficient way of comparing the performances of two or more classifiers.

The most common way to represent ROC in a scalar value is finding the area under the ROC curve (AUC) [65]. ROC curves are widely used in signal detection theory to illustrate the

tradeoff between hit rates and false alarm rates of a classifier [66, 67]. As AUC is the area under a unit square in ROC, its value lies between 0 and 1. The worst classifier which classifies all positive or all negative has an AUC of 0.5. Hence, no realistic classifier should have an AUC below 0.5 [63]. An ideal classifier has an AUC value equal to 1. Figure 2.13 explains three classifiers represented as {A, B, C}. In this figure, C represents the worst classifier with an AUC value of 0.5 followed by a better classifier in B and eventually A being the best classifier with an ideal AUC value of 1.

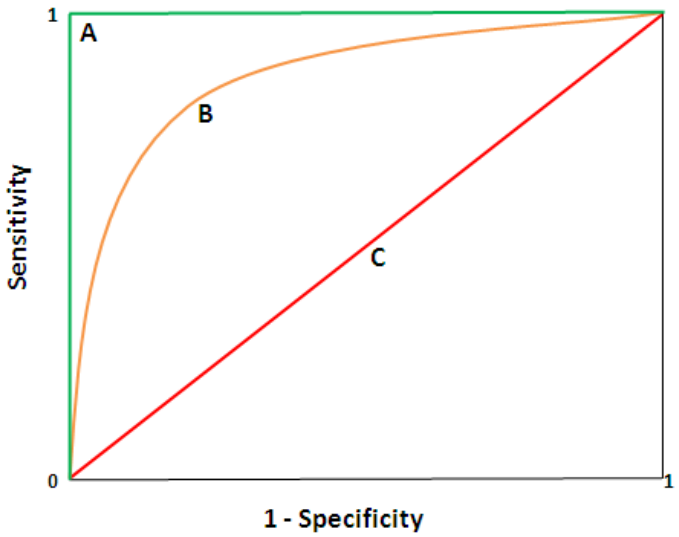


Figure 2.13: Diagram showing a receiver operating characteristics

ROC is used only in the case of binary classifiers where the output class is one among two of the trained classes. Multiclass classification problem is the classification problem where more than two classes needs to be classified in the output stage for a multiple input single output classifier system. In order to compare the performance of multi class classifiers, a robust classification algorithm based on probability estimation trees is proposed [68].

3

FUZZY RULE BASED CLASSIFICATION SYSTEM FOR TRANSIENT IDENTIFICATION

The present chapter elucidates on the analysis undertaken in using FRBCS for classifying some of the transients from the steam water system in a nuclear power plant. The importance of proper feature selection is elaborately explained by analyzing the effects on the performance of the classifier by conducting experiments with different input variables to the FRBCS. This chapter also clarifies the importance of response time in analyzing the performance of FRBCS in online transient classification of system which have very short cycle time.

3.1 Introduction

The performance of modeling a system using fuzzy modeling more often depends on interpretability and accuracy of the FRBS. These two phenomena are contradictory in nature where interpretability indicates the capability to express the behavior of the real system in a understandable way, and accuracy defines the capability to faithfully represent the real system [69]. The usual reasoning follows the ‘‘principle of incompatibility” formulated by Zadeh [70]: *As the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance become almost mutually exclusive characteristics. The closer one looks at a real-world problem, the fuzzier becomes its solution*. It is usually assumed that a highly complex

systems lack interpretability. Based on the requirement, fuzzy modeling is segregated into linguistic fuzzy modeling approach and piecewise fuzzy modeling approach. Linguistic fuzzy modeling approach primarily focuses on the interpretability and then tries to improve the accuracy [71]. Piecewise fuzzy modeling approach primarily focuses on the accuracy and then tries to improve the interpretability [72]. Fuzzy modeling not only aims at a high interpretability but also a better accuracy. Therefore, recent research emphasizes on good accuracy-interpretability trade-off in order to attain a compact and robust fuzzy system.

FRBCS as stated earlier, is a category of FRBS dedicated to classification problems, is widely recognized for its robustness to imperfect data and interpretability [73]. The accuracy of a FRBCS is calculated by comparing the fuzzy model output and the actual output. The percentage of the correctly classified fuzzy output from the total dataset gives the accuracy of the model. The interpretability of a FRBCS depends on the selection of a proper input feature set. A input feature set comprises of the model structure, number of input variables, number of rules, number of linguistic terms, shape of the membership function and so on [74]. The selection of an appropriate input feature set depends on the expert knowledge as there is no predefined procedure to do so.

NPP having a cycle time of 200 milliseconds, generates about ten thousand signal data each cycle. There is a need of developing a highly interpretable FRBCS for online transient identification in NPP which has such a short cycle time. To make a highly interpretable system, the complexity must be reduced but this may not result in desired accuracy. On the other hand, any increase in the input parameters of the FRBCS, i.e., increase in complexity may increase the accuracy of the classifier but makes the model low in interpretability. The present chapter explains the process of development of a FRBCS for transient identification in the PFBR. It also

explains about the importance of feature selection for quick and better classification results. In addition to this, the present chapter also stresses on some of the salient features that needs to be taken utmost care before concluding on the final classification result from the developed online FRBCS.

3.2 FRBCS for transient identification

Modeling a classifier using fuzzy logic can be handy as well as efficient as a redundant prediction system in order to aid the operator to take legitimate decisions. The identification of transients at the earliest without any ambiguity helps the operator in taking necessary steps in order to combat the severity of the event. The thesis attempts at developing a prediction system whose quick and unambiguous output guides the controllers to take necessary actions. Ideally a system should be robust to external interferences to provide best results. Filtering out the noise data by pre-processing it clears the chances of interference. Apart from this any other interference does not affect the FRBCS. Based on the time taken by the transient from the initiation of the event till the reactor is shutdown, the transients are categorized into fast occurring transients and slow occurring transients. Fast occurring transients such as transients in the neutronics system, lead to reactor shutdown within a few scanning cycles at the earliest. Hence, monitoring such kind of transients is not possible as the reaction time is too short. On the contrary, the slow occurring transients such as the transients in the steam water system give ample time for the classifier to predict the correct transient. The sooner the transient is predicted correctly by the classifier, the more time is provided to the operator to react to it. Hence, some of the transients in the steam water system are considered for the current experimental purpose. The experimental data for each considered transient are collected from the operator training simulator (OTS). The OTS is a full scope replica type simulator which simulates various plant conditions

and operations. It is used to train the operator extensively on monitoring and reacting to various transients and malfunction in the plant as well as to prepare the operator for real plant scenarios. Noise is not generated in OTS intentionally or unintentionally, hence the implication of it is nearly weak in FRBCS. The current study is conducted in two phases. The first phase consisted of three transients and the second phase consisted of five transients from the steam water system as the considered events. The first phase is again divided into two sub-phases and the second phase is divided into three sub-phases where each sub-phase consist of a designated set of input features along with its membership function and rule base. All these different sets of simulated experiments generated from the OTS are conducted to generalize the final inference without much of ambiguity.

3.3 Phase-1 of FRBCS

In this phase, three transients from the steam water system are considered for event classification using FRBCS. These transients are as follows

i. One Condensate Extraction Pump Trip with stand by not taking over (1-CEP trip)

ii. Both the Condensate Extraction Pumps Trip (2-CEP trip)

iii. Both Boiler Feed Pumps Trip (2-BFP trip)

A detailed description about the PFBR plant is reported in Chetal et al. [3]. In order to analyze the effect of selection of the input features, two diverse plant parameters are chosen for experimental purpose. These plant parameters are reactor inlet temperature (θ_{RI}) and deaerator level (DL).

- ***Reactor Inlet Temperature***

Reactor inlet temperature (θ_{RI}) in PFBR is the temperature of the primary sodium at the inlet to the core. At full power condition of the reactor, this temperature is 397°C.This temperature may increase when any abnormality occurs in the reactor. The alarm comes when this temperature increases by 5°C and the reactor is tripped when it increases by 10°C. Considering the transients, θ_{RI} gets affected a little late compared to many other nearby plant parameters. It is a trip signal which initiates reactor shutdown when the value crosses the threshold. Figure 3.1 shows the time series pattern of θ_{RI} for the considered transients.

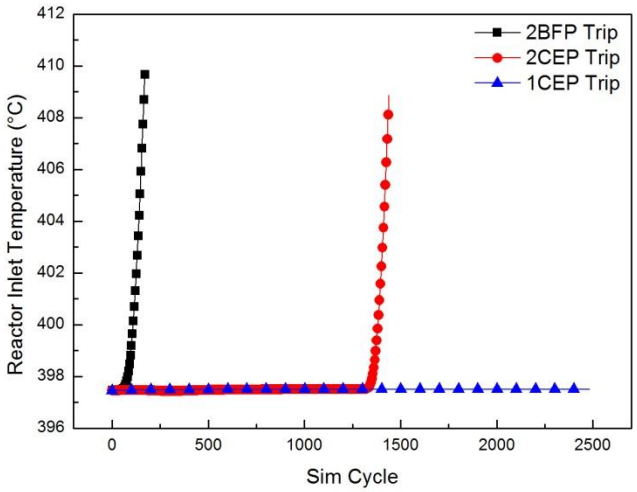


Figure 3.1: Time series pattern of θ_{RI} for the considered transients

- ***Deaerator Level***

Deaerator is used for the removal of dissolved gases from feed water which is then fed to the once through Steam Generator for heat absorption from the secondary sodium (refer Fig 1.1). In the deaerator, the deaerator level (DL) has to be maintained at 2.4m with the pressure and temperature being around 485kPa and 150°C respectively. The feed water from here is pumped into the steam generator (SG) by the boiler feed pumps. The feed water temperature in the

deaerator is maintained at 150°C to meet the steam-water system start up condition. This enables thermal recycling in order to maintain the system efficiency. DL gets immediately affected on the occurrence of the considered transients. Figure 3.2 depicts the time series pattern of DL for the considered transients.

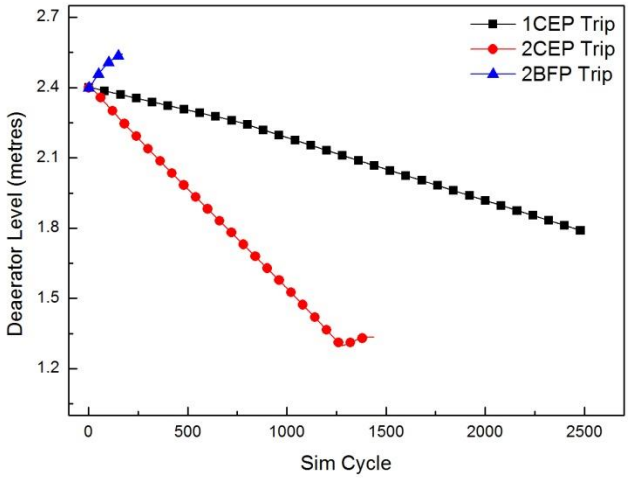


Figure 3.2: Time series pattern of DL for the considered transients

3.3.1 FRBCS for Phase–1 using θ_{RI} and $\Delta\theta_{RI}$ (Phase–1a)

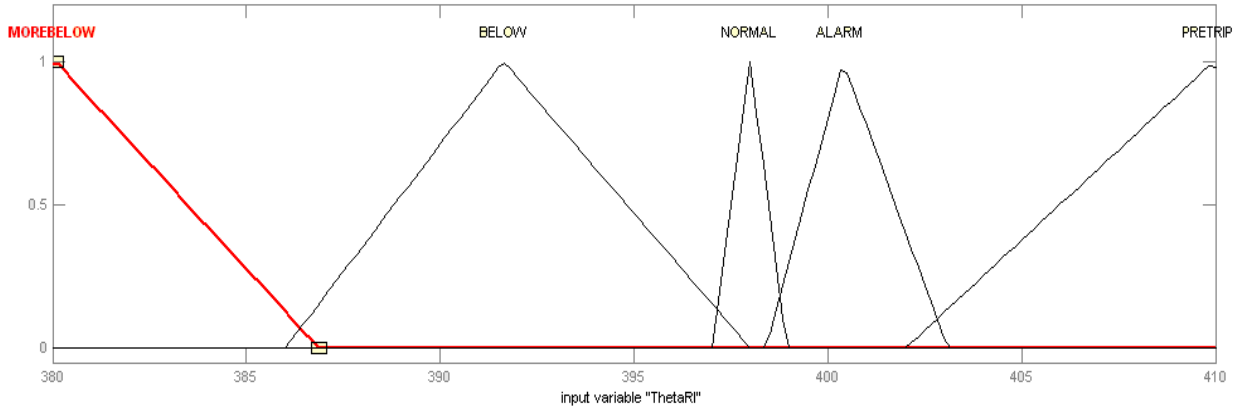
The OTS generates data on all plant parameters every 200 milliseconds and θ_{RI} is collected from it. The change in reactor inlet temperature ($\Delta\theta_{RI}$) is considered as another input parameter along with θ_{RI} and is calculated by subtracting two consecutive θ_{RI} values. $\Delta\theta_{RI}$ along with θ_{RI} would provide more clarity to the transient prediction both being the features of the input feature vector. Based on the prepared input feature vector and the experts' opinion, the fuzzy linguistic variables for the input and the output domains are segregated.

3.3.1.1 Membership functions for Phase-1a

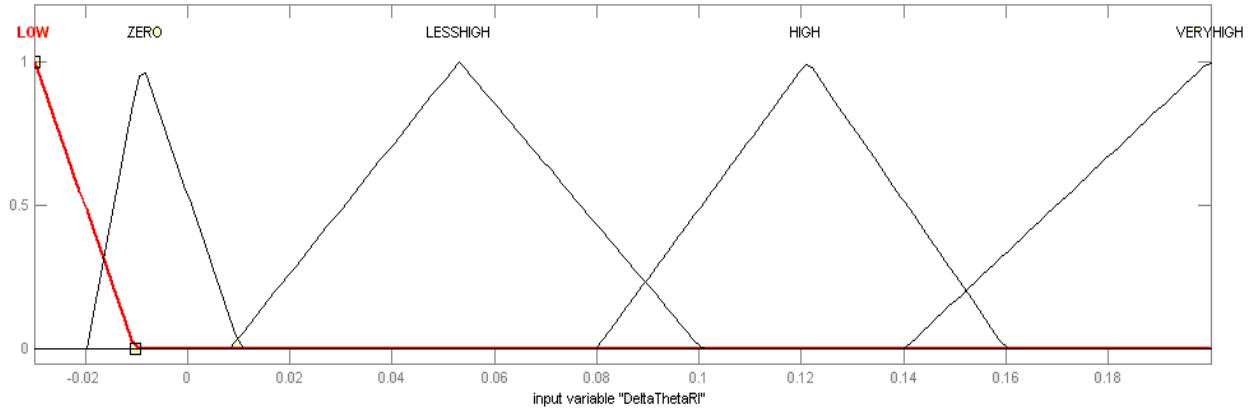
The input membership functions and the output membership functions are prepared based on the domain segregation using experts' knowledge as shown in Fig. 3.3. This segregation

involves the design threshold for each input parameter which is decided by the designers along with some intuition regarding the number of windows so that the rule base does not become over crowded. A handshaking between the threshold values and a deep analysis on the pattern of each parameter for each class along with some logic initiates a faster approach to finding the final output. This segregation also keeps in mind the unique combination of the input parameters which provides a clear classification boundary of different classes that is to be classified. This approach is the same for all the cases in this chapter. FP abbreviates for full power which means the plant running in full power condition. The classification of an event is done based on the defuzzified value. If the defuzzified output is nearly 1 then the classified event is 1CEP trip, if it is around 2 then 1or2CEP trip and so on as shown in Fig. 3.3(c). This analogy is adhered throughout this thesis for FRBCS.

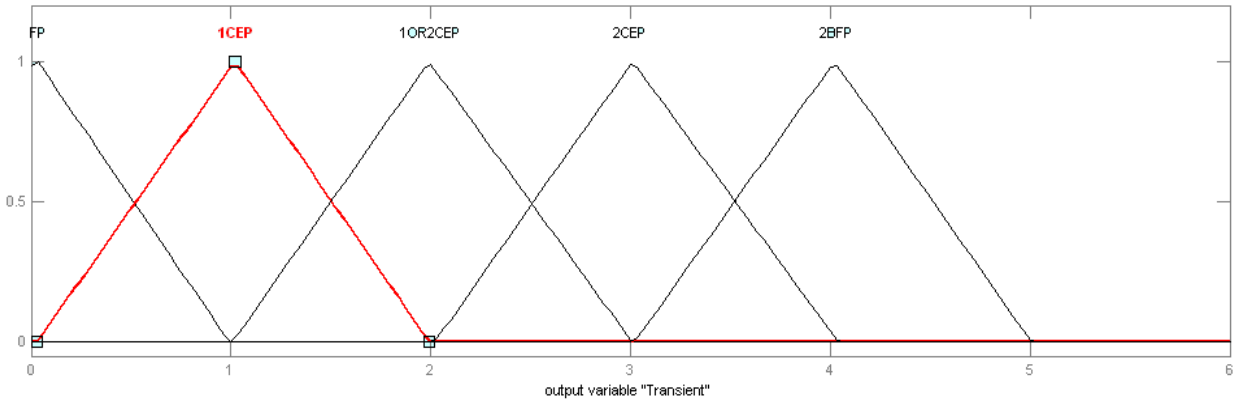
a. Fuzzy values for the input linguistic variable θ_{RI} = {MOREBELOW, BELOW, NORMAL, ALARM, PRETRIP}
b. Fuzzy values for the input linguistic variable $\Delta\theta_{RI}$ = {LOW, ZERO, LESSHIGH, HIGH, VERYHIGH}
c. Fuzzy values for the output linguistic variable = {FP, 1-CEP, 1OR2CEP, 2-CEP, 2-BFP}



(a)



(b)



(c)

Figure 3.3: Membership functions of FRBCS for Phase-1a

3.3.1.2 Rule Base for Phase-1a

The preparation of the rule base commences once the generation of the input membership functions and the output membership function terminates. The rule base consists of simple "IF-

THEN" statements that cover the entire domain of the considered transients. The preparation of the rule base is a very important stage and requires experts' suggestion to produce the best one. The rule base produced for Phase–1a is shown in Table 3.1. The empty boxes in the tables denote that, under any circumstances, the data would not fall in this category for the considered transients. A fuzzy output variable as 1or2-CEP Trip is considered here because for a certain period of time, it is not conclusive on the occurrence of either 1-CEP Trip or 2-CEP Trip as the values are quite identical. Some of the examples of the rules generated by the rule base are as follows.

If θ_{RI} is NORMAL and $\Delta\theta_{RI}$ is ZERO then the plant is running on FP.

If θ_{RI} is MOREBELOW and $\Delta\theta_{RI}$ is LOW then 1-CEP trip has happened.

If θ_{RI} is BELOW and $\Delta\theta_{RI}$ is LOW then 1or2-CEP trip has happened.

If θ_{RI} is BELOW and $\Delta\theta_{RI}$ is LESSHIGH then 2-CEP trip has happened.

If θ_{RI} is NORMAL and $\Delta\theta_{RI}$ is LESSHIGH then 2-BFP trip has happened.

Table 3.1: Rule base of FRBCS for Phase –1a

θ_{RI} / $\Delta\theta_{RI}$	LOW	ZERO	LESSHIGH	HIGH	VERYHIGH
MOREBELOW	1-CEP	1-CEP	-	-	-
BELOW	1OR2-CEP	1OR2-CEP	2-CEP	2-CEP	2-CEP
NORMAL	-	FP	2-BFP	2-CEP	2-CEP
ALARM	-	2-BFP	2-BFP	2-BFP	2-CEP
PRETRIP	-	-	-	2-CEP	2-CEP

3.3.1.3 Output for Phase-1a

Phase-1a produces the output class as the transient which is identified using the FRBCS model. This output is based on the crisp output produced by the FRBCS using the output

membership function described earlier. The decision boundary which governs the output of the FRBCS is fixed at a range of –0.5 and +0.5 of the mean value of the output membership function. This segregation remains same for all the further consideration of output result for a FRBCS.

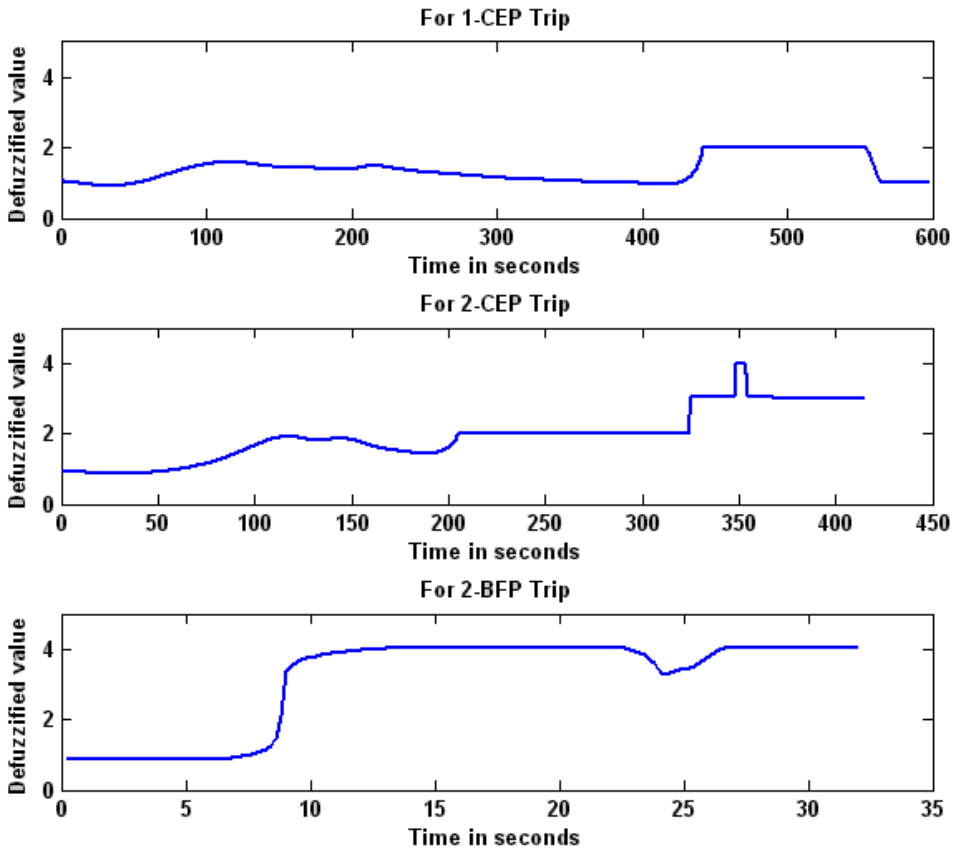


Figure 3.4: Defuzzified output of FRBCS for Phase-1a

3.3.1.4 Response time for Phase-1a

The time taken by a FRBCS in order to give a confident result on the occurrence of an event is called the response time. A graph is considered stable when there is a very minimal fluctuation in the defuzzified output value for an event classification. Based on this criterion, the response time for all the phases have been found out. Some figures show very stable graphs which is due to the parameter chosen. Others are not so stable which is due to the incapability of

the parameter to classify properly. Figure 3.4 shows that the defuzzified output for 1-CEP trip and 2-CEP trip remains the same for a considerable amount of time till about 200s. This shows that the trend for the considered input feature, i.e., θ_{RI} is same for both these transients. The defuzzified output for 2-CEP trip is 3 at around 330s which is its value according to the output membership function and stabilizes around 350s. Hence the response time of the FRBCS for Phase-1a for 2-CEP trip is 350s. Regarding, 1-CEP trip, the θ_{RI} responds in a similar manner like 2-CEP trip for some period initially. Once the confusing period between 450s to 550s is over where it is unsure of a 1-CEP or 2-CEP trip, there is a conclusive evidence of the occurrence of 1-CEP trip with a defuzzified value of 1 as in the output membership function in Fig 3.3c. It is also evident from this figure that the defuzzified output for 1-CEP trip stabilizes at 1 after 550s, hence that being its response time. Compared to both these transients, 2-BFP trip for this case attains stability by 12s which signifies that the response time for this transient is comparatively better than the other two considered transients. Beyond 35s, the plant would have reached the shutdown state. The fluctuations in all the graphs in this chapter is due to the safety actions being taken by the system automatically in the OTS in order to counter act these abnormal changes in the plant.

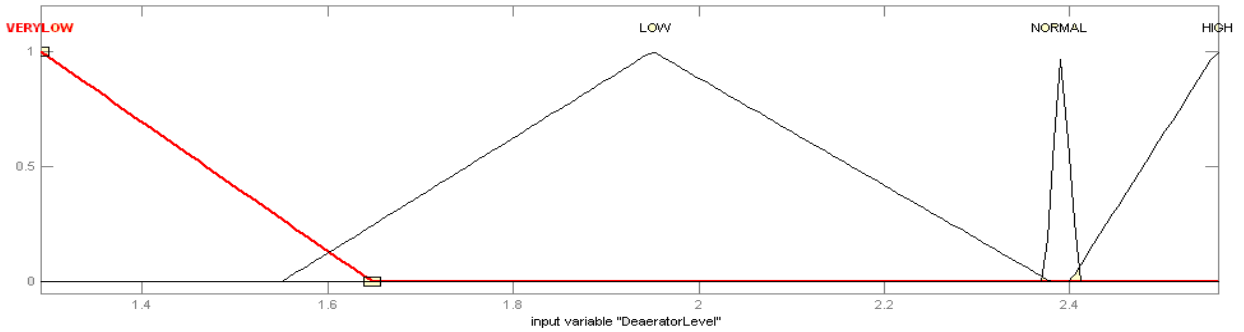
3.3.2 FRBCS for Phase–1 using DL and ΔDL (Phase–1b)

Phase-1b considered the parameters which is very near to the occurrence of the event and gets most affected. Including DL, the change in deaerator level (ΔDL) promises to give an insight on the characteristic of each transient which eventually might result in better classification accuracy and response time.

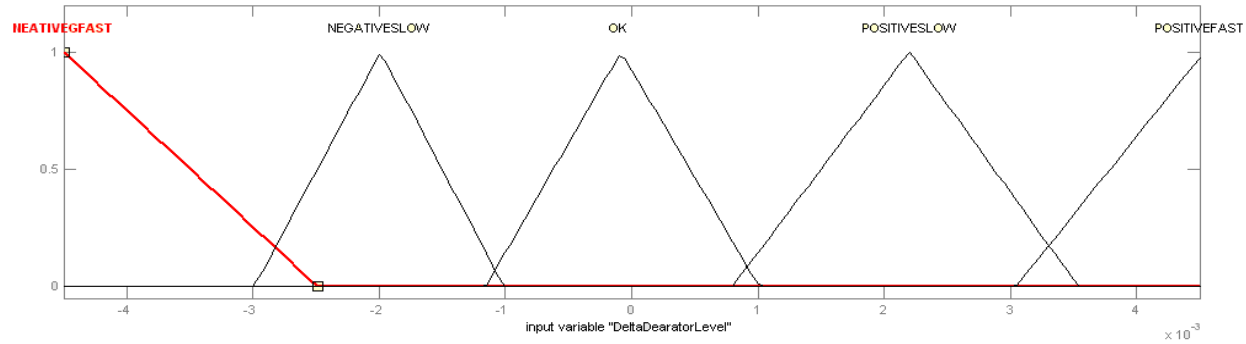
3.3.2.1 Membership functions for Phase-1b

Similar to Phase-1a, the input and output membership functions are prepared based on the segregation of the selected input feature domain, i.e. DL and ΔDL, into specific linguistic fuzzy variables. It is observed in Fig. 3.5c, that the generation of an intermediate transient generated in Phase-1a, i.e., 1or2CEP trip is avoided in this case on the usage on DL and ΔDL as the input features. This is because DL and ΔDL parameters have clear segregation in the values avoiding any sort of confusion related to the events that need to be classified. This means at no sampling time, the DL and ΔDL values were same for 1-CEP and 2-CEP trip, thus giving clear boundaries for classification which was missing in Phase-1a. The segregation of the input and output domains into fuzzy linguistic variables is done as the following.

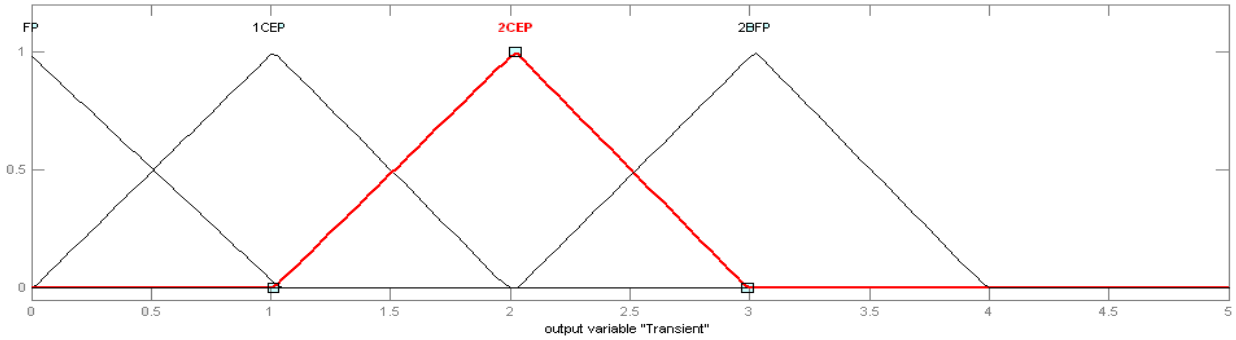
a. Fuzzy values for the input linguistic variable DL = {VERYLOW, LOW, NORMAL, HIGH}
b. Fuzzy values for the input linguistic variable ΔDL = {NEGATIVEFAST, NEGATIVESLOW, OK, POSITIVESLOW, POSITIVEFAST}
c. Fuzzy values for the output linguistic variable = {FP, 1-CEP, 2-CEP, 2-BFP}



(a)



(b)



(c)

Figure 3.5: Membership functions of FRBCS for Phase-1b

3.3.2.2 Rule base for Phase-1b

The fuzzy rules governing the fuzzy linguistic variables of DL and ΔDL are generated and stored as the rule base for Phase-1b referred as Table 3.2. This approach is similar to the rule base preparation in Phase-1a. Some of the fuzzy rules for this sub-phase are as follows.

If DL is NORMAL and ΔDL is OK then the plant is running on FP.

If DL is LOW and ΔDL_I is NEGATIVESLOW then 1-CEP Trip has happened.

If DL is VERYLOW and ΔDL is NEGATIVEFAST then 2-CEP Trip has happened.

If DL is HIGH and ΔDL is POSITIVEFAST then 2-BFP Trip has happened.

Table 3.2: Rule base of FRBCS for Phase-1b

DL	ΔDL	NEGATIVE FAST	NEGATIVE SLOW	OK	POSITIVE SLOW	POSITIVE FAST
VERYLOW		2-CEP	2-CEP	2-CEP	2-CEP	-
LOW		2-CEP	1-CEP	1-CEP	1-CEP	2-BFP
NORMAL		-	1-CEP	FP	-	2-BFP
HIGH		-	-	-	2-BFP	2-BFP

3.3.2.3 Output for Phase-1b

The procedure in finding the defuzzified output of the FRBCS for Phase-1b is similar to that of Phase-1a. Each data is fed as input one by one and based on the input membership function the rule is selected from the rule base which produces the fuzzy output. This fuzzy output is defuzzified using the output membership function which yield the crisp output. The defuzzified or crisp outputs for the three considered transients are shown in Fig. 3.6.

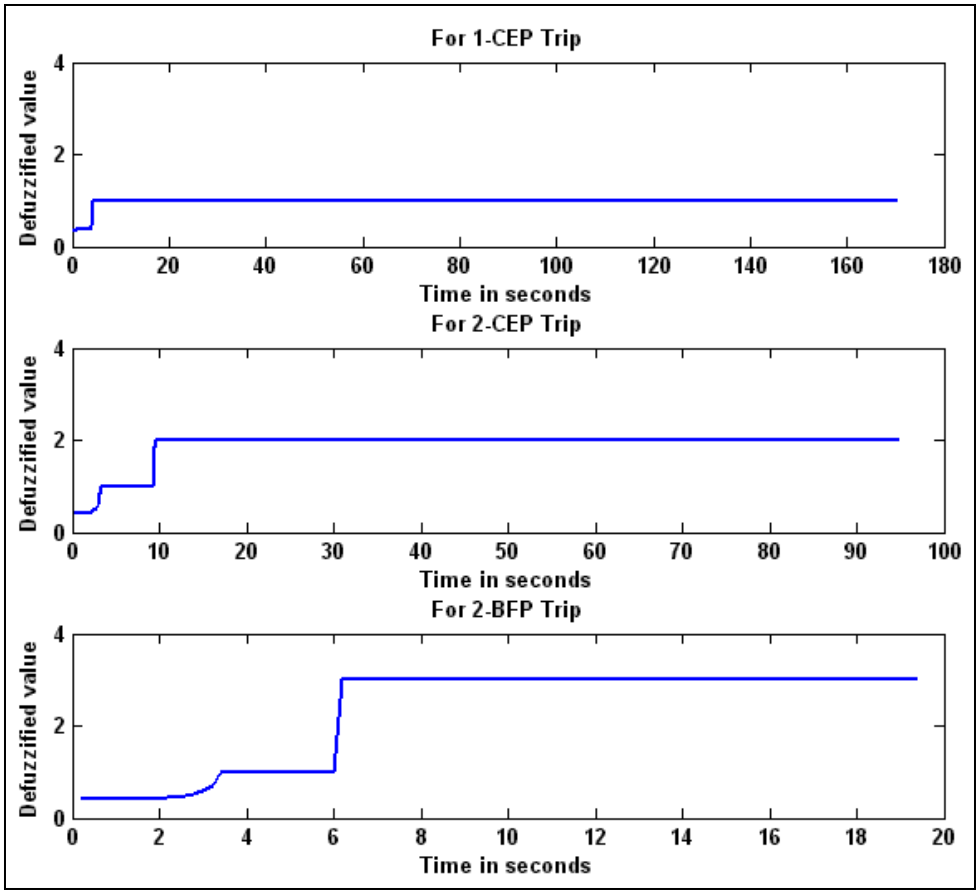


Figure 3.6: Defuzzified output of FRBCS for Phase-1b

3.3.2.4 Response time for Phase-1b

Figure 3.6 clearly depicts that for initial 10s, 1CEP trip and 2-CEP trip produces same defuzzified output from the FRBCS for Phase-1b. It is only after 10s, the defuzzified output changes to 2 and stabilizes at this value which confirms that the transient occurred is a 2-CEP trip. The defuzzified output for 1-CEP trip remains constant throughout the series after this moment. Hence, it is learnt from the figure that the response time for 1-CEP trip is around 15s whereas for 2-CEP trip it is around 12s. This 3s window is kept to bypass the noise interference if any. A similar window is there in each case in this chapter. The response time for 2-BFP trip is 8s as evident from Fig. 3.6.

3.3.3 Comparison between Phase-1a and Phase-1b

The response time for each transient in both the phases has been mentioned in Table 3.3. The difference in the response time for 2-BFP trip is quite less comparing the difference in case of 1-CEP trip and 2-CEP trip. This shows that Phase-1a which uses DL and ΔDL as the input features produces better result in all the three transients compared to the FRBCS in Phase-1b which uses θ_{RI} and $\Delta\theta_{RI}$ as their input features. This concludes that the parameter near the occurrence of the transient gives far better result as they get affected at the earliest compared to the trip parameter which is far away from the occurrence of the event.

Table 3.3: Response time of the two FRBCS in Phase-1

Phase	Inputs	1-CEP Trip	2-CEP Trip	2-BFP Trip
1a	θ_{RI} -$\Delta\theta_{RI}$	550 sec	350 sec	12 sec
1b	DL-ΔDL	15 sec	12 sec	8 sec

3.4 Phase-2 of FRBCS

In this phase, five transients from the steam water system are considered compared to three transients in Phase-2, for event classification using FRBCS in order to increase the complexity of the transients. These transients are as follows

i. One Condensate Extraction Pump Trip with stand by not taking over (1-CEP trip)

ii. Both the Condensate Extraction Pumps Trip (2-CEP trip)

iii. One Boiler Feed Pump Trip with stand by not taking over (1-BFP trip)

iv. Both Boiler Feed Pumps Trip (2-BFP trip)

v. Both Cooling Water Pumps Trip (2-CWP trip)

As inferred in Phase-1, the parameters nearer to the occurrence of the event produces better result compared to the far away parameters. In order to analyze the effect of selection of the input features which are nearer to the occurrence of the events, two such plant parameters are chosen for experimental purpose. These plant parameters are deaerator level (DL) and condensate pressure (CP). DL is taken because it gave the better result of the two in the previous case and has been explained earlier in the present chapter. Figure 3.7 shows the time series pattern of DL for the considered transients.

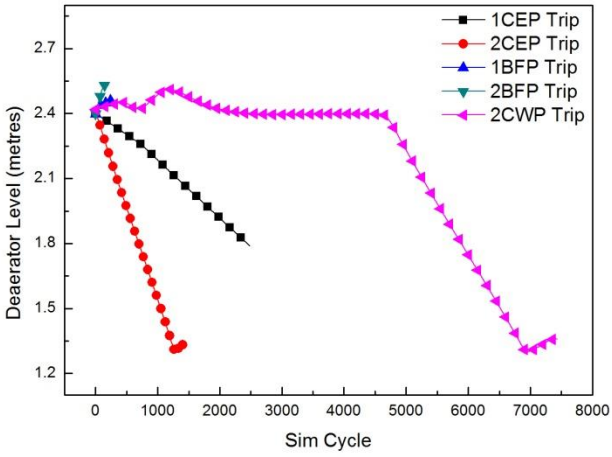


Figure 3.7: Time series pattern of DL for the considered transients

- ***Condensate pressure***

After the super heated steam has passed the last stage of the turbine, the leftover heat content in the steam which is approximately 750MWt has to be dumped. This is done in the condenser with the help of sea water. In the condenser, heat exchange takes place where the steam flows through the shell side and the sea water flows through the tube side. Here, the condenser pressure or otherwise called as condensate pressure is maintained at 9.2kPa by the

help of air evacuation system with two vacuum pumps. If this pressure crosses the designed limits then that may affect the blades of the last stage of the turbine. Figure 3.8 depicts the time series pattern of CP for the considered transients.

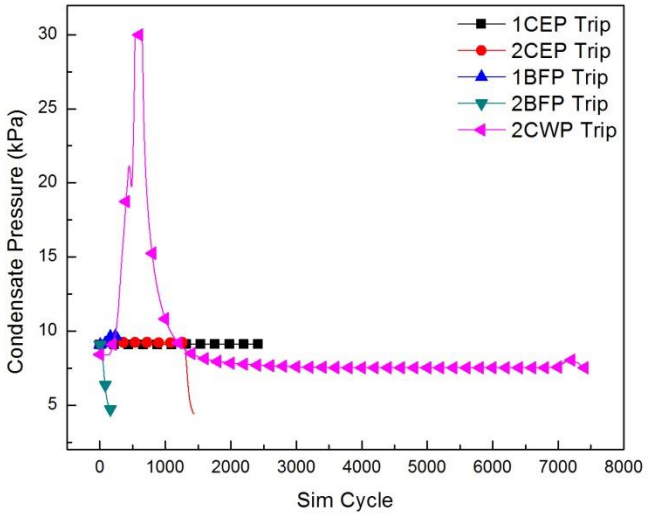


Figure 3.8: Time series pattern of CP for the considered transients

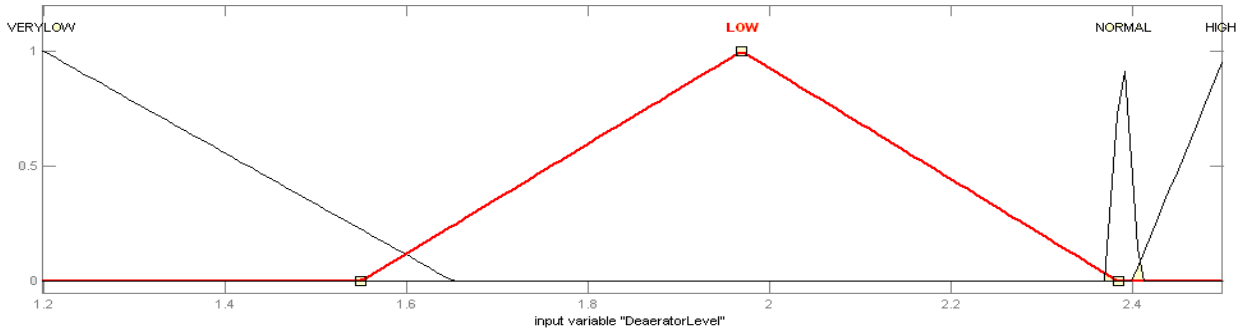
3.4.1 FRBCS for Phase–2 using DL and ΔDL (Phase–2a)

Phase-2a has considered change in DL along with the DL as the input feature for the FRBCS. It is similar to Phase-1b with increased number of classes which somehow complicates the system.

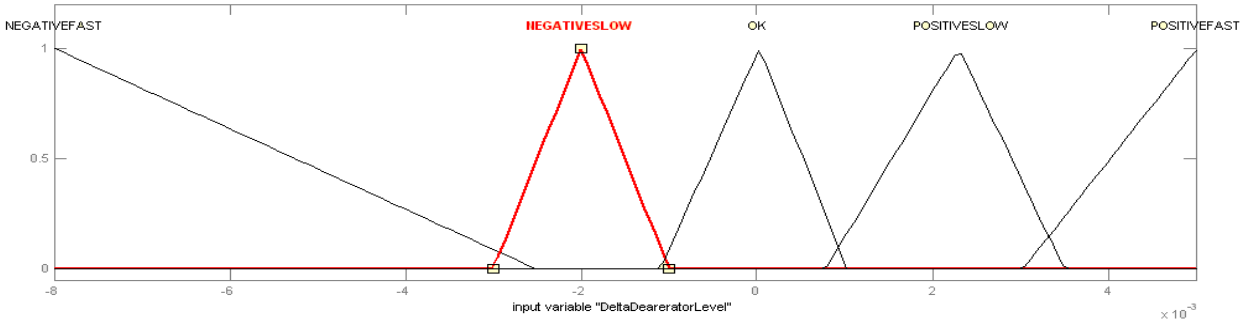
3.4.1.1 Membership functions for Phase-2a

Similar to the previous phase, the input membership function and the output membership function are prepared considering the domain of the selected input features in Phase-2a as depicted in Fig. 3.9. Phase-2a has same input membership function as that in Phase-1b whereas the output membership function has introduced two new transients. The segregation of the input and output domains into fuzzy linguistic variables is done as the following.

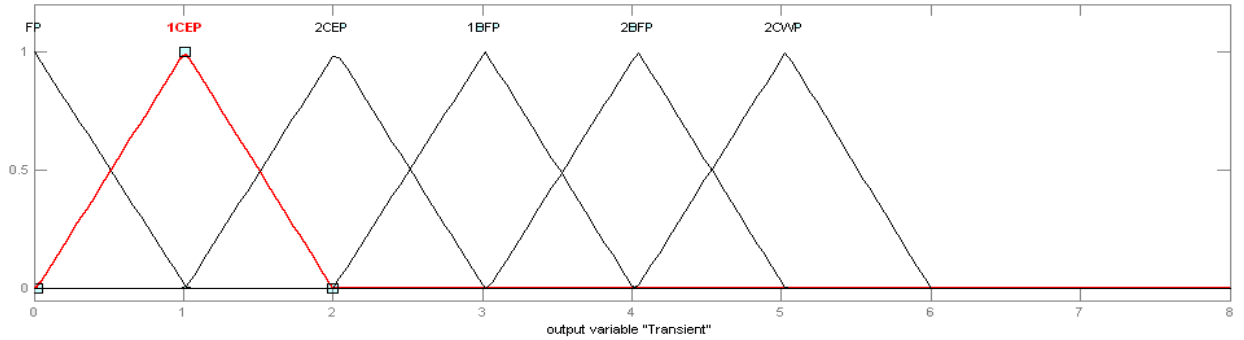
a. Fuzzy values for the input linguistic variable DL = {VERYLOW, LOW, NORMAL, HIGH}
b. Fuzzy values for the input linguistic variable ΔDL = {NEGFAST, NEGSLOW, OK, POSSLOW, POSFAST}
c. Fuzzy values for the output linguistic variable = {FP, 1-CEP, 2-CEP, 1-BFP, 2-BFP, 2-CWP}



(a)



(b)



(c)

Figure 3.9 : Defuzzified output of FRBCS for Phase-1b

3.4.1.2 Rule base for Phase-2a

The rule base is generated by producing fuzzy rules for each linguistic fuzzy variable shown in Table 3.4. Some of the examples of the generated rules are as follows.

If DL is NORMAL and ΔDL is OK then the plant is running on FP.

If DL is LOW and ΔDL is NEGATIVE SLOW then 1-CEP Trip has happened.

If DL is VERY LOW and ΔDL is NEGATIVE SLOW then 2-CEP Trip has happened.

If DL is HIGH and ΔDL is OK then 1-BFP Trip has happened.

If DL is HIGH and ΔDL is POSITIVE SLOW then it is 2BFP Trip has happened.

If DL is HIGH and ΔDL is NEGATIVE FAST then it is 2CWP Trip has happened.

Table 3.4: Rule base of FRBCS for Phase-2a

DL	ΔDL	NEGATIVE FAST	NEGATIVE SLOW	OK	POSITIVE SLOW	POSITIVE FAST
VERYLOW		2-CEP	2-CEP	2-CWP	2-CEP	-
LOW		2-CEP	1-CEP	1-CEP	2-BFP	2-BFP
NORMAL		-	1-CEP	FP	1-BFP	2-BFP
HIGH		2-CWP	2-CWP	1-BFP	2-BFP	2-BFP

3.4.1.3 Output for Phase-2a

The defuzzified output in this phase is based on the assumption as undertaken in Phase-1. Figure 3.10 shows the defuzzified output during online event classification of transients.

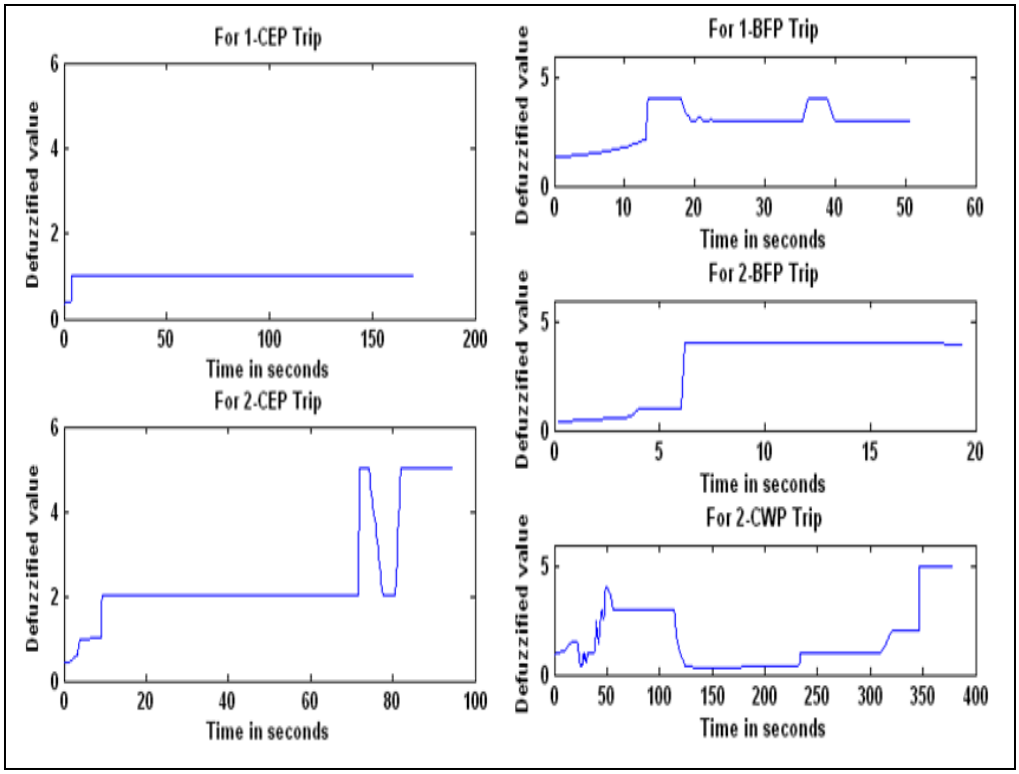


Figure 3.10: Defuzzified output of FRBCS for Phase-2a

3.4.1.4 Response time for Phase-2a

As mentioned earlier in Phase-1, the data collected for 1-CEP trip and 2-CEP trip have almost similar data for certain period of time. Hence, it is advisable to wait for certain duration in order to reach at any conclusion on the occurrence of the transient. Figure 3.10 show that till 10s both these transients show same defuzzified output. Hence, only after this duration, conclusive evidence is gathered on to the correct classification of transient. Hence, the response time for 1-CEP trip is 20s as the defuzzified output stabilizes at 1 beyond this mark. Similarly, the defuzzified output for 2-CEP trip stabilizes at 2 beyond a response time of 15s. Beyond 70s, the 2CEP trip shows spikes due to automatic safety measures taken by the controllers in the plant. These spikes enlighten that safety actions are taken at this point of time after the occurrence of the event where as these actions could have been taken well before by the help of FRBCS. For 1-BFP trip, the output stabilizes at nearly 3 with a response time of 20s whereas for 2-BFP trip

the output stabilizes at nearly 4 within 10s. For 2-CWP trip, the response time is 360s. This is because the variance of DL and ΔDL is not of considerable amount in order to classify the occurrence of 2-CWP trip. If the diagnosis of 2-CWP trip was stopped at 200 or 300 seconds, event classification would not be an accurate one. As at this time, according to the defuzzified output value it shows some other event and not 2-CWP trip. Hence, at any instance, accurate event classification is not possible before the response time**.**

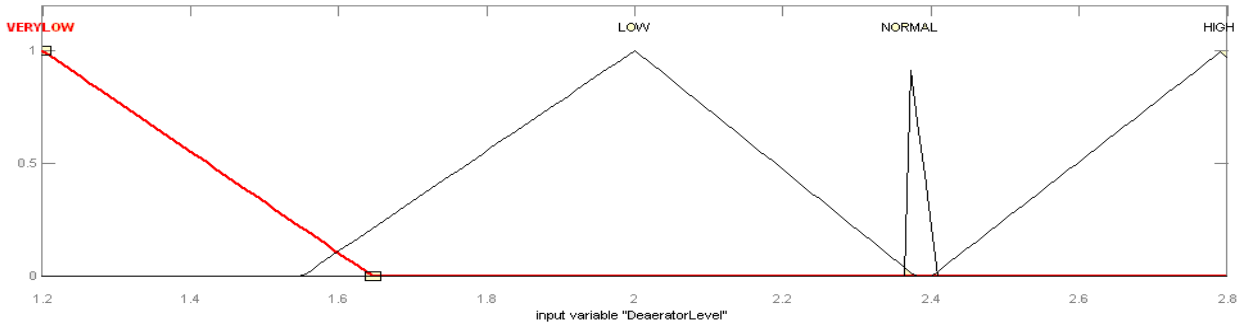
3.4.2 FRBCS for Phase–2 using DL and CP (Phase–2b)

Phase-2a for classification of 2-CWP trip produced a response time on the higher side. From previous result, it is quite evident to try with some other nearby parameter which would presumably have a shorter response time than the previous case. Hence, CP is used to formulate a new FRBCS along with the DL.

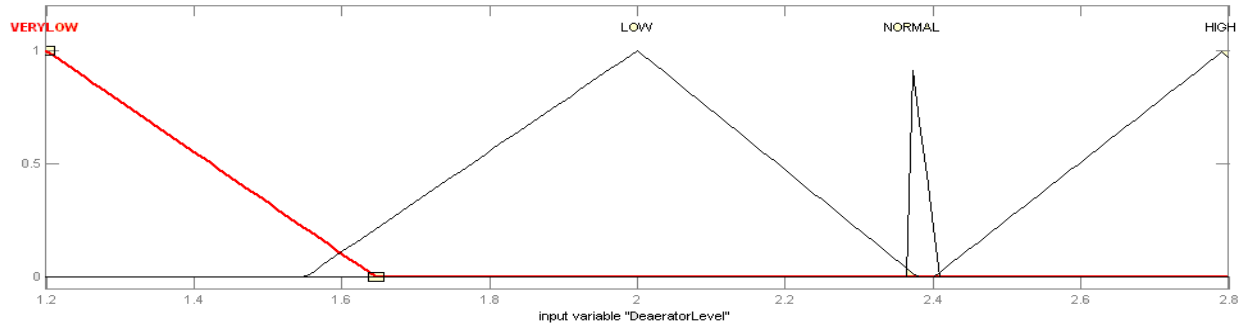
3.4.2.1 Membership functions for Phase-2b

The membership is again segregated based on experience and experts' opinion. The segregation of the fuzzy domain of the input and output data is done using fuzzy linguistic variables as shown in Fig. 3.11. These fuzzy linguistic variables are as follows.

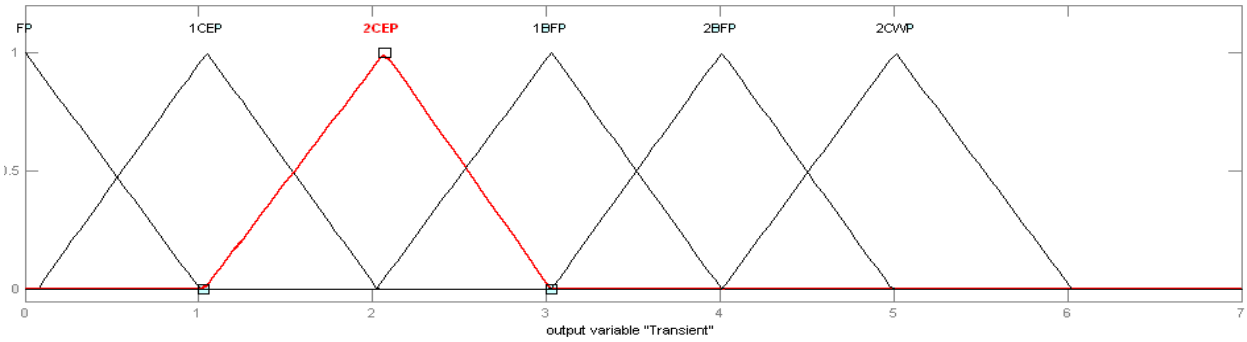
a. Fuzzy values for the input linguistic variable DL = {VERY LOW, LOW, NORMAL, HIGH}
b. Fuzzy values for the input linguistic variable CP = {LESS, OK, LITTLEMORE, MORE, HUGE}
c. Fuzzy values for the output linguistic variable = {FP, 1-CEP, 2-CEP, 1-BFP, 2-BFP, 2-CWP}



(a)



(b)



(c)

Figure 3.11: Membership function of FRBCD for Phase-2b

3.4.2.2 Rule base for Phase-2b

The fuzzy rules governing the linguistic variables of DL and CP are generated and stored in their respective rule base as shown in Table 3.5. For example,

If DL is NORMAL and CP is OK then the plant is running on FP.

If DL is LOW and CP is LESS then 1-CEP trip has happened.

If DL is VERYLOW and CP is LESS then 2-CEP trip has happened.

If DL is HIGH and CP is LITTLEMORE then 1-BFP trip has happened.

If DL is HIGH and CP is LESS then it is 2BFP trip has happened.

If DL is HIGH and CP is HUGE then it is 2CWP trip has happened.

Table 3.5: Rule base of FRBCS for Phase-2b

DL	CP	LESS	OK	LITTLE MORE	MORE	HUGE
VERYLOW		2-CEP	2-CEP	2-CEP	-	-
LOW		2-CEP	1-CEP	2-CEP	2-CEP	2-CEP
NORMAL		-	-	FP	-	1-BFP
HIGH		2-CWP	2-BFP	1-BFP	1-BFP	1-BFP

3.4.2.3 Output for Phase-2b

The defuzzified output of Phase-2b is illustrated in Fig. 3.12.

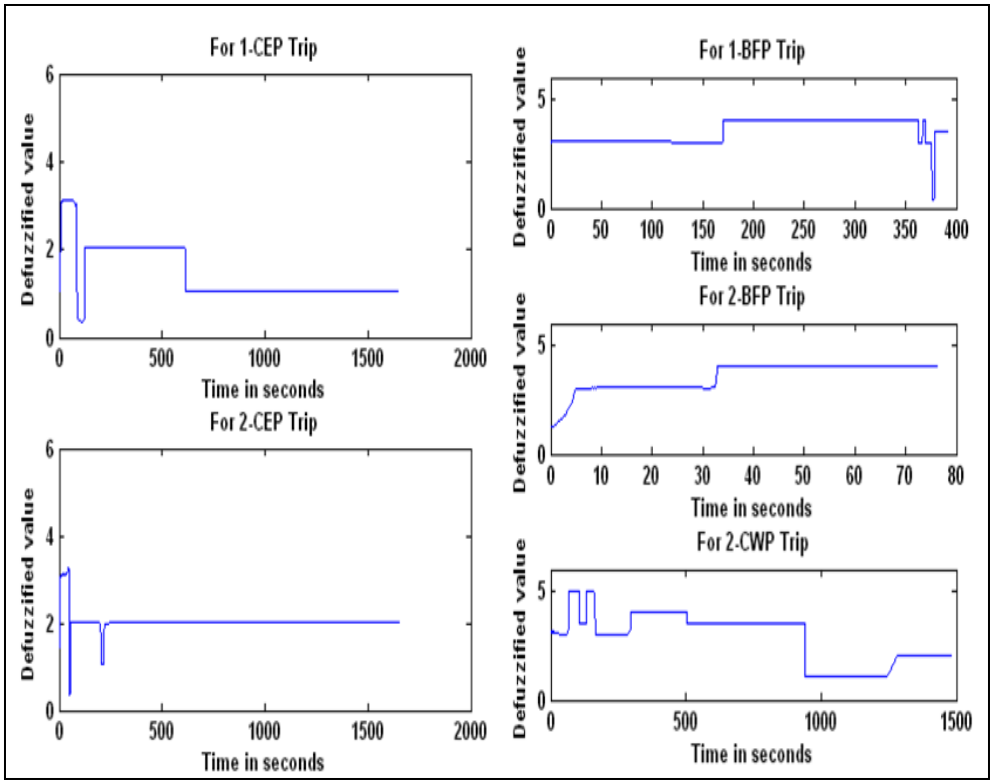


Figure 3.12: Defuzzified output of FRBCS for Phase-2b

3.4.2.4 Response time for Phase-2b

From Fig. 3.12, it is observed that the defuzzified values for 1-CEP trip and 2-CEP trip are almost same i.e. nearly 2, till 600s. Afterwards, the values changed to 1 for 1-CEP trip and remained at 2 for 2-CEP trip. This means that the response time for 1-CEP trip is around 600s and for 2-CEP trip it is 610s. Similar is the case for 1-BFP trip and 2-BFP trip where the defuzzified value is 3 till 35s. Afterwards, the output changed to 4 for 2-BFP trip and remained at 3 for 1-BFP trip for certain duration. 1-BFP trip reaches output 4 after about 170s due to the values of the input parameters, viz., Deaerator level and Condensate pressure reaching to values nearly equal to that of 2-BFP trip case. Due to this the defuzzified output touches 4 after 170 seconds. This is a possible scenario provided the controllers are near perfect ones. The response time is 50s and 40s for 1-BFP trip and 2-BFP trip respectively. From Fig. 3.12, it is also

observed that a defuzzified value of 5 is seen in 2-CWP trip within 70s. During this time itself, the system is confident on the occurrence of CWP trip. According to the data, on occurrence of 2-CWP trip, the CP reaches up to 30kPa. No other considered events have a CP of 30kPa under any circumstances.

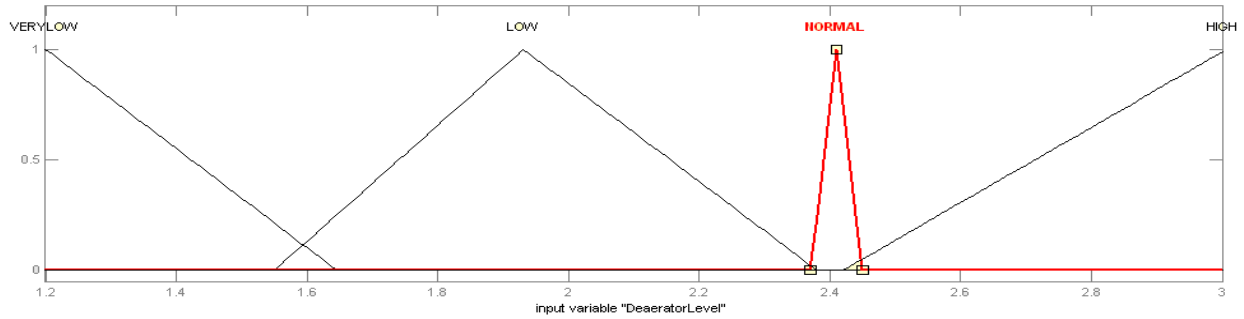
3.4.3 FRBCS for Phase–2 using DL, ΔDL and CP (Phase–2c)

In this case, three parameters are considered instead of two as input variables to a FRBCS. These parameters are accounted based on the advantages from the previous approaches for developing FRBCS.

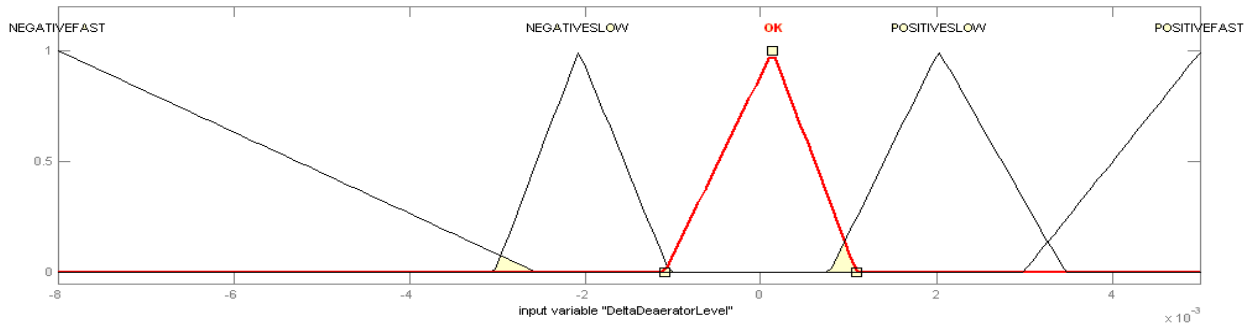
3.4.3.1 Membership functions for Phase-2c

Depending on data collected from the transient based on the three selected input variables, the membership functions are prepared as shown in Fig. 3.13. The three parameters are from Phase-2a and Phase-2b, i.e., DL, ΔDL and CP as the combination of these presumed to give better result considering the benefits of both the sub-phases individually.

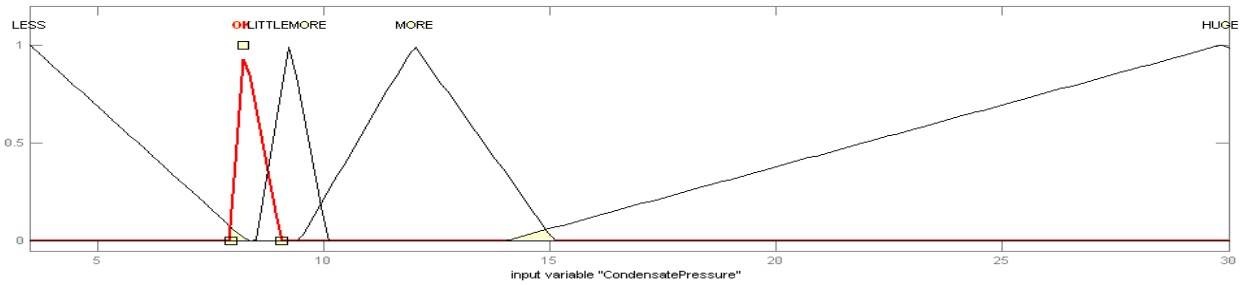
a. Fuzzy values for the input linguistic variable DL = {VERYLOW, LOW, NORMAL, HIGH}
b. Fuzzy values for the input linguistic variable ΔDL = {NEGATIVEFAST, NEGATIVESLOW, OK, POSITIVESLOW, POSITIVEFAST}
c. Fuzzy values for the input linguistic variable CP = {LESS, OK, LITTLEMORE, MORE, HUGE}
d. Fuzzy values for the output linguistic variable = {FP, 1-CEP, 2-CEP, 1-BFP, 2-BFP, 2-CWP}



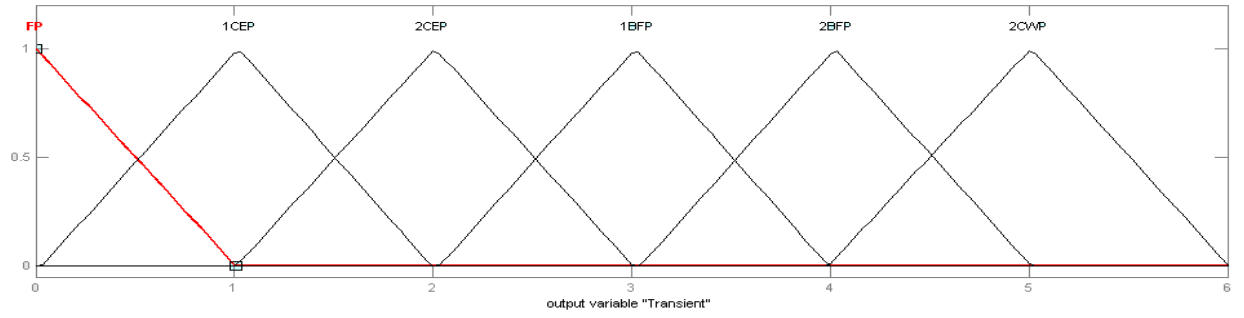
(a)



(b)



(c)



(d)

Figure 3.13: Membership functions of FRBCS for Phase-2c

3.4.3.2 Rule base for Phase-2c

To prepare the rule base with three parameters on a two dimensional matrix, some notations are used as shown in Table 3.6. For example say, 'a_1' where 'a' represents DL and the subscript 1 represents the first fuzzy linguistic variable of DL, i.e., very low. Similar is the case for b and c which represent ΔDL and CP respectively. So when it is represented as 'a_1b_1', this represents DL being very low and ΔDL being negative fast.

Table 3.6: Rule base of FRBCS for Phase-2c

ab \ c	c_1	c_2	c_3	c_4	c_5
a_1b_1	2-CEP	2-CEP	-	-	-
a_1b_2	2-CEP	2-CEP	-	-	-
a_1b_3	2-CEP	2-CEP	-	-	-
a_1b_4	-	-	-	-	-
a_1b_5	-	-	-	-	-
a_2b_1	-	-	-	-	-
a_2b_2	-	-	-	-	-
a_2b_3	1-CEP	1-CEP	1-CEP	1-CEP	1-CEP
a_2b_4	1-CEP	-	-	-	-
a_2b_5	-	-	-	-	-
a_3b_1	-	-	-	-	-
a_3b_2	-	-	-	-	-
a_3b_3	1-BFP	Full Power	1-CEP	2CWP	2CWP
a_3b_4	-	2-BFP	-	-	-
a_3b_5	-	-	-	-	-
a_4b_1	-	-	-	-	-
a_4b_2	-	-	-	-	-
a_4b_3	1-BFP	1-BFP	1-BFP	-	2-CWP
a_4b_4	2-BFP	2-BFP	-	-	-
a_4b_5	-	-	-	-	-

3.4.3.3 Output for Phase-2c

The defuzzified output of the FRBCS for Phase-2c is represented in Fig. 3.14.

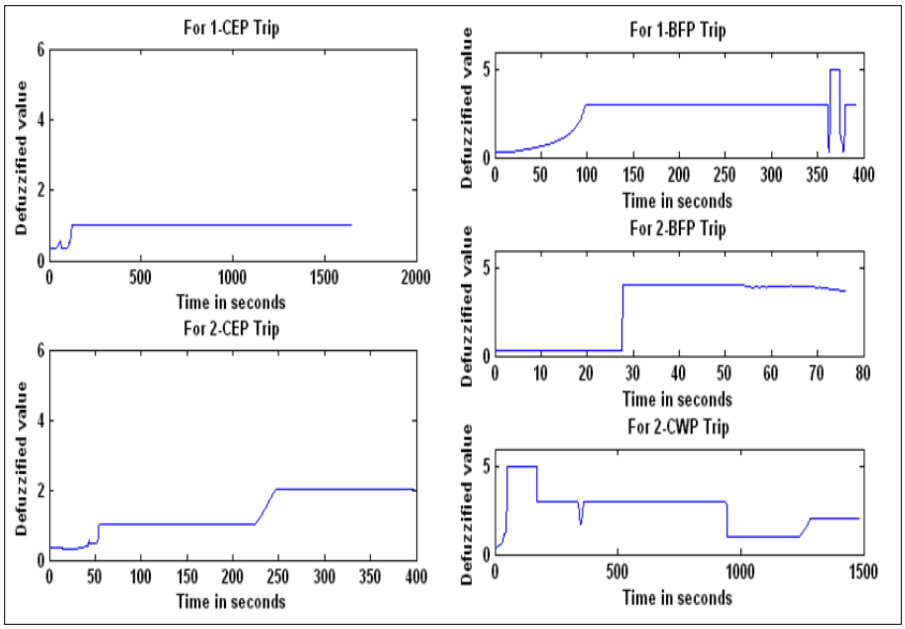


Figure 3.14: Defuzzified output of FRBCS for Phase-2c

3.4.3.4 Response time for Phase-2c

Figure 3.14 shows that the defuzzified outputs for Phase-2c are quite static with low fluctuations in the initial stages. Here also, there is need to wait a considerable duration in order to distinguish between 1-CEP Trip and 2-CEP Trip as the data are almost equal, as mentioned earlier. For 1-CEP Trip, this system takes around 260s and the defuzzified output stabilizes at 1 whereas for 1-CEP Trip it takes around 250s and stabilizes at 2. The response time for 1-BFP Trip is found to be around 100s as it stabilizes at 3 thereafter. 2-BFP Trip is classified quicker than others in this system, around 30s, at a stabilized output value of 4. As the defuzzified value-5 which belongs to 2-CWP trip never arose in any of the other four transient cases, an output for such a short interval of time with defuzzified value-5 was assumed to be a 2-CWP trip. Admittedly, this is not the most convincing way of classifying this event and work is in progress at this centre to achieve to a more appealing and appreciable algorithm. Another reason for such a result is due the improper selection of the input features for the classification of this event. A shorter period of time would certainly make the conclusions more prominent. Hence, the

response time for 2-CWP trip as it reaches the defuzzified output value of 5 is found to be 55s. It stabilizes there for certain duration. Further change in the output in case of 2-CWP Trip is due to the automatic safety measures been taken up by the controllers.

3.4.4 Comparison between response times of the three FRBCS in Phase-2

Based on the OTS experiments conducted on the above mentioned three sub-phases in Phase-2, Table 3.7 shows the comparison based on the response time of each FRBCS towards the specific input variables in this phase. Phase-2a with input variables DL-ΔDL to the FRBCS produces output with considerably shorter response time in four of the five events compared to FRBCS for Phase-2b and Phase-2c. As the response time of 2-CWP trip in FRBCS for Phase-2a is more, a change in input variable from ΔDL to CP is chosen. This is considered because 2-CWP trip has a direct impact on CP. This change of input variable reduces the response time for 2-CWP trip but in return increases the response time of all other events. This is because, the CP is not able to classify the other four events along with DL as these four transients have no immediate impact on CP. Taking all these observations into consideration, FRBCS for Phase-2c is produced with three input variables, i.e., DL-ΔDL-CP. This system gives a result which turned out to be a trade-off between the FRBCS of Phase-2a and Phase-2b. The response time of the FRBCS for Phase-2c is in the midst of both the previous phases. Finally, it is found that FRBCS for Phase-2a is better for the first four events with a response time in a reasonable low range and FRBCS for Phase-2c being better for classifying 2-CWP trip event only. Hence, DL-ΔDL variables proved to be mostly a better combination of input variables for the selected event classification.

Table 3.7: Response time of the three FRBCS in Phase-2

Phase	Input Variables	1-CEP	2-CEP	1-BFP	2-BFP	2-CWP
2a	DL-ΔDL	**20 sec**	**15 sec**	**20 sec**	**10 sec**	360 sec
2b	DL-CP	600 sec	610 sec	50 sec	40 sec	70 sec
2c	DL-ΔDL-CP	260 sec	250 sec	100 sec	30 sec	**55 sec**

3.5 Summary

The experiment conducted to analyze the effect of the FRBCS for online transient classification inferred that a classifier performs well with an appropriate set of input variables.

- The present chapter infers that parameters near to the occurrence of the transient have a major effect on classifying the correct event.
- Phase-1 demonstrates that FRBCS with DL and ΔDL as input variables proves to be efficient with a shorter response time than θ_{RI} and $\Delta\theta_{RI}$.
- With an increase in the number of transients to be classified, FRBCS with DL and ΔDL as input variables has a shorter response time than FRBCS with DL and CP as input variables.
- Increasing the number of input variables considering FRBCS with DL, ΔDL and CP reduces the response time of 2-CWP Trip event only. All the other events produce a higher response time comparing the response time of FRBCS with DL and ΔDL as input variables.
- It is not always the increase in the number of input variables which favors more accurate pattern recognition. Judiciously going for the feature selection for a fuzzy system definitely leads to a more accurate system whose interpretability is not penalized much.
- The appropriate process for online classification of events, whose system's scanning cycle time is too small, is by monitoring the output from a classifier system for a certain number of

cycles and then deciding the actual occurrence of an event which gives conclusive evidence for the final outcome.

- This system can be a redundant one in case of any annunciation failures and avoids digging into the log records in order to conclude on the occurrence of a transient. It eventually reduces information overloading on the operator.

4

DEVELOPMENT OF TRAINING DATASET REDUCTION FOR TRANSIENT IDENTIFICATION

The present chapter outlines the significance of a novel approach of reduction in training samples referred as training dataset reduction (TDR). The TDR algorithm has been delineated and the performance of the classifier based on TDR algorithm is investigated on some real world datasets. Further, the practicability of TDR algorithm, based on the previous results, is validated on datasets collected from the transients of a sodium cooled fast reactor (SFR).

4.1 Introduction

In supervised machine learning algorithms for classification problem related quandary, training the classifier with the training data and testing the same using the test data is the universal approach. Quite often these training data and test data are selected randomly from the full dataset in the ratio of 80%:20% or 70%:30% respectively. These percentages have a momentous role when the full dataset is huge demanding a large memory and the classifier taking unacceptably long training time. The computational complexity rises exponentially in using an efficient machine learning algorithm for big data [75]. On completion of the training procedure, the accuracy of the classifier network with the test data might not be satisfactory demanding repetition of the procedure with additional modifications. This process often becomes tedious and cumbersome. This unacceptable accuracy stems from the quality of the dataset

considered. In such case, it is required to reduce the number of samples in the training dataset so that the unnecessary samples are dropped. This technique removes samples in two groups. One among them is removing samples that are far from the class centroid. Using outlier samples or support vector samples via fuzzy entropy for SVM [76] or multi category proximal SVM [77] are some of these approaches. Another approach is removing samples near to each class centroid. K-means clustering technique [78] is used to achieve such kind of reduction in samples. The approach undertaken in this chapter is inspired from the concept of removing samples far from the class centroid using a different approach and is termed as training dataset reduction (TDR) approach. Reducing the training data also reduces the time complexity of the classifier[79].

In any classification related problem, the endeavor of the algorithm is to accomplish the best accuracy possible for the classifier network. Multi kernel learning (MKL) is an approach which improves classification performance effectively but with a large computational complexity. Advancement to MKL is the multi-kernel classification machine based on the Nystrom approximation which has a better average recognition and a short training time [80]. Another area of application where higher accuracy is of immense necessity is the classification of the occurrence of transient in a safety critical environment like a NPP where the classifier response should be quick, accurate and unambiguous in order to avoid any lethal accidents. A valuable approach to this matter is the practice of fuzzy rule base system for transient classification in NPP as described in Chapter 2.

With a number of supervised machine learning algorithms available, the selection of the best one for classification purpose of a dataset may vary based on the characteristics of the learning algorithm [81, 82]. This analysis becomes imperative when some of the performance parameters of the learning algorithm are of higher precedence compared to others. A real world

example which supplements this statement is the accuracy of the classifier network in case of a NPP which is of highest priority compared to its training time and computational complexity. Hence for such situations, the performance comparison of various supervised machine learning algorithms is necessary to find the best one with maximum accuracy.

The TDR approach is a preprocessing step carried out before training the classifier network. In this approach, the training dataset is reduced and the new reduced training dataset is fed as the input to the classifier network. Beyond this step, the classifier network is made to learn using a chosen supervised machine learning algorithm. This approach is fruitful only if the performance of the classifier network does not deteriorate much on undergoing this approach. The TDR approach selects some of the samples which would be used to train the classifier network based on a particular cut-off Euclidean distance. In this chapter, some of the methods used to find out this cut-off Euclidean distance is described which finally leads to the preparation of a reduced training dataset known as the new training dataset. The feasibility of TDR approach is verified using some real world datasets based on the performance of the classifier network. Furthermore, a diverse dataset considering transients in a SFR for classification is investigated for verification of the TDR approach.

4.2 Concept of Training Dataset Reduction

TDR is a novel approach to reduce the number of samples in the training dataset which is fed as input to a classifier. It is assumed that the size of the full dataset (FD) is R x C, where R represents the total number of rows or samples and C represents the total number of columns or features. The output class of a classifier is always a function of the feature vector of a sample and the weight vector of the classifier at that instant. Generally, a classification process undergoes two phases, training phase and test phase and for each of these phases, the full dataset is

segregated into training dataset and test dataset respectively. Prior to the commencement of the training phase, normalization of the full dataset followed by TDR is carried out which produces a new training dataset. In the training phase, this new training dataset is fed as input to a classifier which gets trained based on a supervised machine learning algorithm. The class of each training sample is compared with its real class and the error is back propagated to change the weights of the classifier in order to reduce the error in the subsequent iteration. This process is continued till the system converges, i.e., either the error or the number of epochs reaches its respective thresholds. Finally, after the classifier is trained, the test phase commences with the testing dataset used to validate the trained classifier. Generally, the accuracy of the classifier is calculated based on the number of correct test classifications with respect to the total number of test samples. It is known that that

$$\boldsymbol{Output\ Class = f(feature\ vector\ , weight\ vector)} \tag{38}$$

4.2.1 Normalization

The full dataset is initially normalized before undergoing the process of TDR. Normalization is a process of transforming a data to a specific range, say between 0 and 1. This part is necessary in order to avoid a huge difference in the values of the features which might otherwise make the system ill-conditioned. This process more often assures stable convergence of weights and biases.

$$\boldsymbol{x_{normalized} = \frac{x - minx}{maxx - minx}} \tag{39}$$

4.2.2 Steps involved in TDR

I. The given normalized dataset is segregated into 80% of training dataset with *N* rows, and the rest 20% of test dataset with *M* rows and each row containing *C* columns representing the features as shown in Fig. 4.1.

II. From the test dataset, *A*% of *M* rows are randomly selected such that at least one test data from each unique class from the full dataset is picked. If the above condition does not satisfy then this step is repeatedly carried out with increased percentage of *A*. This selected test dataset is called as pickup test (pTe*)* dataset which has *m* rows.

III. For the i^{th} row present in the pickup test dataset, the Euclidean distance (ED) is calculated with respect to the k^{th} row of the training dataset where *k* varies from 1, 2, . . . ,*N*. This results in a column matrix $ED_i = [ED_{i1}, ED_{i2}, \ldots, ED_{ik}, \ldots, ED_{iN}]^T$ where

$$\boldsymbol{ED_{ik}(pTe_i, Tr_k) = \sqrt{\sum_{j=1}^{C} (Tr_{k,j} - pTe_{i,j})^2}} \quad \textbf{(40)}$$

Certainly, due to the curse of dimensionality, Gaussian based functions can be used as distance metrics rather than Euclidean distance for higher dimensional datasets. In this thesis, a maximum of 10 features has been considered. So, finding the distance using Euclidean distance is justified for such low dimensional datasets. Hence, Euclidean distance is chosen as the distance metrics in this thesis.

IV. The ED_i is then sorted in an ascending order of their distance values.

V. The selection of the corresponding training dataset is done using the sorted Euclidean distance matrix based on $ED_{cut\text{-}off}$. $ED_{cut\text{-}off}$ is assumed as the radius of a hypothetical circle which segregates the selected samples from the non selected ones in the entire

domain represented as a rectangle in Fig. 4.2. A pickup test data sample is placed at the centre of this hypothetical circle. The training sample whose ED_i is less than $ED_{cut\text{-}off}$ is selected. $ED_{cut\text{-}off}$ is calculated using either of the following methods.

i. Mean – αStandard deviation method

ii. Area selection method

VI. The selected training sample from training dataset is appended to the new training dataset such that redundancy in the selection of the training samples is avoided.

VII. Step III to step VI are repeated for $i = 1, 2, \ldots, m$.

VIII. Now, the new training dataset is used for training the classifier using any machine learning algorithm.

IX. Finally, the test dataset is used to test the classifier's performance.

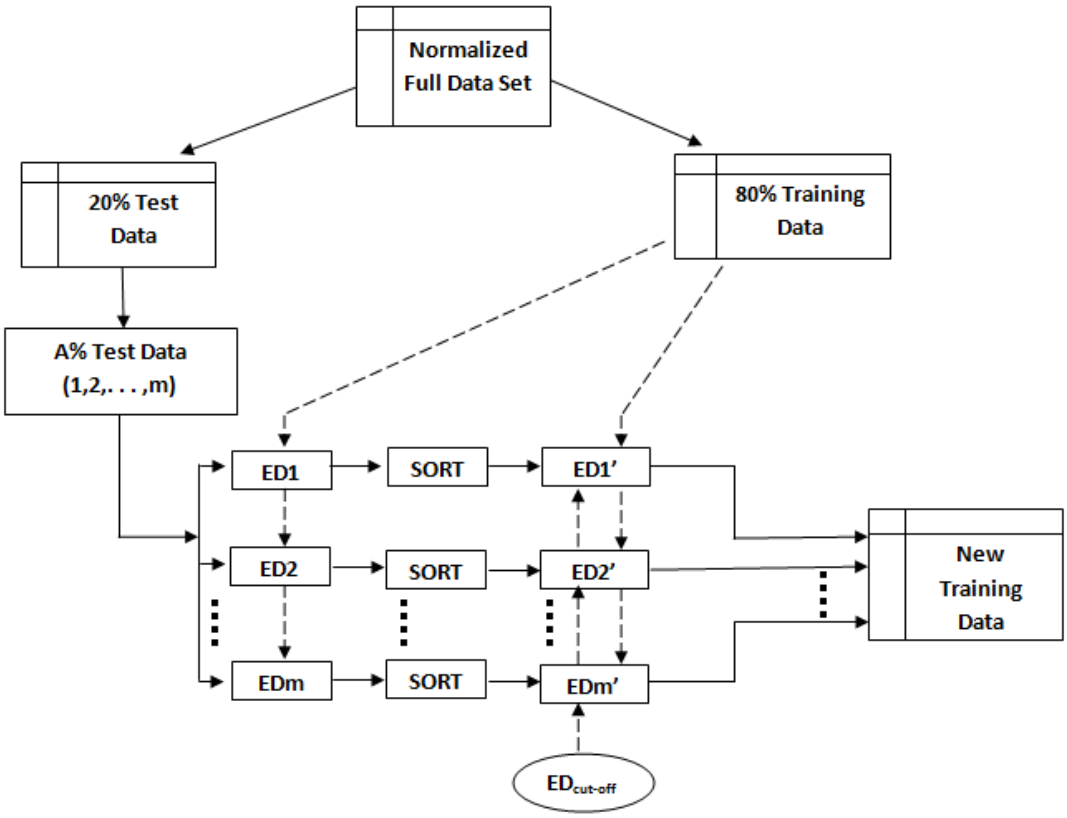


Figure 4.1: Basic block diagram of TDR approach

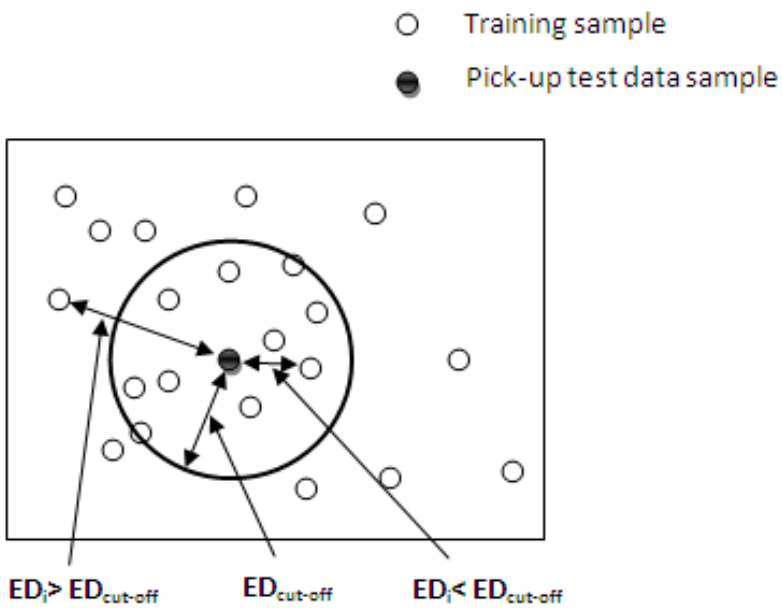


Figure 4.2: Pictorial representation of selecting $ED_{cut\text{-}off}$ in TDR

4.2.3 Mean – αStandard deviation (M-αSD) method

Considering the central limit theorem, it is assumed that the arithmetic mean of the ED values is normally distributed. Figure 4.3 shows the area distribution of the normal distribution curve where the horizontal axis represents the ED ranging from 0 to ∞ and vertical axis is its probability distribution function value. In this method, $ED_{cut\text{-}off}$ is calculated using the mean (μ_{ED}) and standard deviation (σ_{ED}) of the ED vector for each pickup test sample as represented in Eq. 4.4. In Eq. 4.4, α is an integer which preferably ranges from 0 to 3.Those training samples whose corresponding ED satisfies Eq. 4.5 are selected and stored in the new training dataset such that no two samples are repeated.

$$\boldsymbol{ED_{cut-off} = \mu_{ED} - \alpha\sigma_{ED}} \quad \textbf{(4.4)}$$

$$\boldsymbol{ED < ED_{cut-off}} \quad \textbf{(4.5)}$$

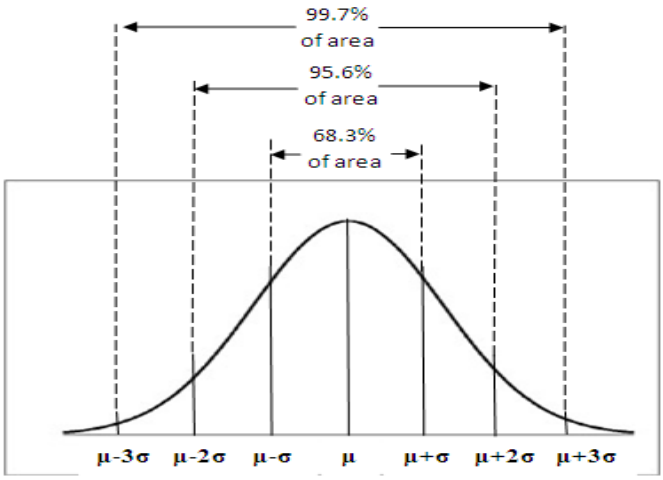


Figure 4.3: Normal distribution Curve

4.2.4 Area selection Method

As mentioned in the previous method, μ_{ED} is normally distributed and integrating its surface within a particular limit gives the area under that limit. Again, it is known that the total area under a normal or Gaussian curve is always equal to 100% or 1. Hence, to calculate the $ED_{cut\text{-}off}$ using this method for a particular range of area covered say 50% or 0.5 from the origin, the limit would vary from 0 to $ED_{cut\text{-}off}$ as shown in Eq. 4.9. In the initial step, the Gaussian distribution values (G_{ED}) are calculated after finding the standardized Euclidean distances (ED_S) using Eq. 4.6 and 4.7.

$$\boldsymbol{ED_s = (ED - \mu_{ED})/\sigma_{ED}} \quad \textbf{(4.6)}$$

$$\boldsymbol{G_{ED} = \frac{1}{\sigma_{ED}\sqrt{2\pi}} \exp\left(\frac{-ED_s^2}{2}\right)} \quad \textbf{(4.7)}$$

Here, the boundary condition for selection of training samples for new training dataset is represented in Eq. 4.8.

$$\boldsymbol{ED_s < ED_{cut-off}} \quad \textbf{(4.8)}$$

The amount of area selected to find its corresponding $ED_{cut\text{-}off}$ is calculated by the following equation.

$$\textbf{Area} = \int_0^{ED_{cut-off}} G_{ED}\ d(ED_s)$$

$$= \int_0^{ED_{cut-off}} \frac{1}{\sigma_{ED}\sqrt{2\pi}} \exp\left(\frac{-ED_s^2}{2}\right) d(ED_s)$$

$$= \frac{1}{\sigma_{ED}\sqrt{2\pi}} \int_0^{ED_{cut-off}} \exp\left(\frac{-ED_s^2}{2}\right) d(ED_s)$$

Let $k = \frac{1}{\sigma_{ED}\sqrt{2\pi}}$ is a constant; then

$$Area = k \int_0^{ED_{cut-off}} \exp\left(\frac{-ED_s^2}{2}\right)\ d(ED_s) = kI \quad \textbf{(4.9)}$$

$$where\ I = \int_0^{ED_{cut-off}} \exp\left(\frac{-ED_s^2}{2}\right)\ d(ED_s) \quad \textbf{(4.10)}$$

The two ways considered in this chapter to integrate Eq. 4.9 in order to calculate the $ED_{cut\text{-}off}$ are analytical integration and numerical integration.

4.2.4.1 Analytical Integration

It is known that on integrating the normal distribution function from 0 to a finite value, say A, an error function is encountered which is defined as shown in Eq. 4.11.

$$\int_0^A e^{-t^2} dt = \frac{\sqrt{\pi}}{2}\ \mathrm{erf}(A) \quad \textbf{(4.11)}$$

$$\Rightarrow \mathrm{erf}(A) = \frac{2}{\sqrt{\pi}} \int_0^A e^{-t^2} dt \quad \textbf{(4.12)}$$

The error function mentioned in Eq. 4.12 can be defined as a Maclaurin series as in Eq. 4.13

$$\mathrm{erf}(A) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n A^{2n+1}}{n!(2n+1)} \quad \textbf{(4.13)}$$

$$= \frac{2}{\sqrt{\pi}} \left(A - \frac{1}{3}A^3 + \frac{1}{10}A^5 - \frac{1}{42}A^7 + \frac{1}{216}A^9 - \ldots\right) \quad \textbf{(4.14)}$$

Substituting the formula of erf(A) in Eq. 4.14 into Eq. 4.11 yields

$$\int_0^A e^{-t^2} dt \; = (A - \frac{1}{3}A^3 + \frac{1}{10}A^5 \; - \frac{1}{42}A^7 + \frac{1}{216}A^9 - \; \ldots) \qquad \textbf{(4.15)}$$

Considering A = $ED_{cut\text{-}off}$ and substituting Eq. 4.15 into Eq. 4.10, results in

$$I = (ED_{cut-off} - \frac{1}{3}ED_{cut-off}^3 + \frac{1}{10}ED_{cut-off}^5 \; - \frac{1}{42}ED_{cut-off}^7 + \cdots) \qquad \textbf{(4.16)}$$

Finally, substituting Eq. 4.16 into Eq. 4.9, results in

$$Area\ selected = \frac{1}{\sigma\sqrt{2\Pi}} \; (ED_{cut-off} - \frac{1}{3}ED_{cut-off}^3 + \frac{1}{10}ED_{cut-off}^5 \; - \frac{1}{42}ED_{cut-off}^7 + \; \ldots) \qquad \textbf{(4.17))}$$

Hence, Eq. 3.17 can be used to calculate the $ED_{cut\text{-}off}$ for a selected area value. But, the problem with this approach is the infinite series and the degree of $ED_{cut\text{-}off}$. Numerical solution to Eq. 3.13 is a tedious task as it is associated with infinite series along with a factorial term.

4.2.4.2 Numerical Integration

Though the numerical integration approach is not accurate, it is relatively simple compared to the analytical integration. Here, the area covered by each adjacent ED_S from left to right is added up till the selected area value is reached. For example, for a selected area of 0.5, the sum of area covered by each ED_S from the previous ED_S is calculated and this process is terminated when the sum of area reaches 0.5. The area of each segment of ED_S and previous ED_S are calculated by assuming each segment to be a trapezium consisting of a rectangle and a right angled triangle. Hence, the area of each trapezoidal segment is the sum of the area of a rectangle and a right angled triangle as highlighted in bold in Fig. 4.4. Here, the ED_S is x_2 with its G_{ED} value of y_2 and the previous ED_S is x_1 with its G_{ED} value of y_1.

Area of the segment = Area of the rectangle + Area of the triangle

$$= (x_2 - x_1)y_1 \; + \frac{1}{2}(x_2 - x_1)(y_2 - y_1)$$

$$= \; (x_2 - x_1)(y_1 \; + \frac{1}{2}(y_2 - y_1))$$

$$= \frac{1}{2}(y_1 + y_2)(x_2 - x_1) \quad (41)$$

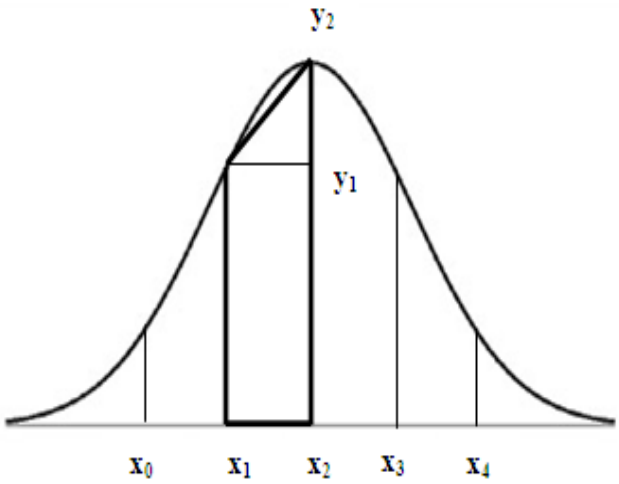


Figure 4.4 : Area selection method for Normal or Gaussian distribution Curve

This approach may not have high precision but computationally it is low in cost. This makes the system really faster compared to using the analytical integration method. On finding the sum of the area nearest to the selected area value, the previous ED_S is represented as the $ED_{cut\text{-}off}$ and the selection of the samples for the new training dataset is carried over using the boundary condition in Eq. 3.8. In the present research, the numerical integration approach is followed for calculating $ED_{cut\text{-}off}$ using area selection method.

4.3 Procedure

In order to check the feasibility of the TDR approach, some of the real world datasets or otherwise known as benchmark datasets are considered as shown in Table 4.1. These datasets are available in the University of California at Irvine (UCI) Repository. These datasets have heterogeneity in their number of samples along with the number of features and classes. Each of these benchmark datasets are fed as input to a classifier network which is trained using a supervised machine learning algorithm. The classifier network used in this approach is a MLP

network which is trained by using the back propagation algorithm in tandem with the gradient descent rule. In the MLP network, only one hidden layer is used with the number of neurons equal to the number of features present in their respective benchmark dataset. The training is carried out in a *k*-fold cross validation manner with *k* equals to 5. This is because the initial segregation of the training and the test data are considered to be 80% and 20% of the FD as shown in Fig. 4.1. In this process, the full dataset is divided into *k* equal number of segments out of which one segment is considered as test data and all the rest segments are considered as training dataset. This approach is repeated for *k*-times where each time the segment assigned as test data is changed without repetition. Towards the end, *k* classification accuracies are collected from the *k*-fold cross validation process and the final classification accuracy of the training algorithm is calculated taking the average of these accuracies.

Table 4.1 describes the considered real world datasets where the column named "previous accuracy" mentions the average classification accuracy of the MLP classifier network for their respective real world dataset without the intervention of the TDR approach. Figure 4.5 shows the block diagram of the training process of the classifier network in a *k*-fold cross validation manner using TDR approach. The average accuracy of the same MLP classifier network which undergoes TDR approach is calculated for the considered real world datasets.

Table 8: Description of Real world datasets

Dataset	#Data	#Features	#Classes	Previous accuracy (%)
Iris	150	4	3	88.7
Wine	188	13	3	96.7
Glass	215	9	6	45.3
WDBC (Wisconsin Diagnostic Breast Cancer)	569	30	2	93.6
Balanced scale	625	4	3	85.4
BCW (Breast Cancer Wisconsin)	684	9	2	96
Wine quality red	1598	11	6	38.5
Wine quality white	4898	11	7	43

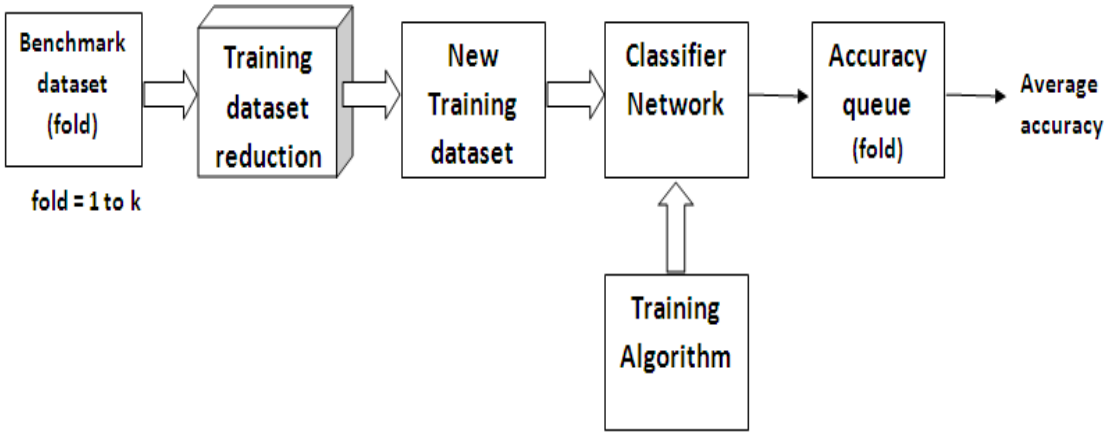


Figure 4.5: Block diagram of training of a benchmark dataset in a k-fold cross validation manner

4.4 Observation

As stated previously in Fig. 4.1, the analysis on the feasibility of the TDR approach depends on the percentage of pickup test data and $ED_{cut\text{-}off}$. In this approach, the values of the percentage of pick-up test data considered for analysis are 5%, 10%, 15% and 20% in order to check their respective effect on the performance of the classifier. These effects are studied using two factors; the percentage reduction in the training samples based on $ED_{cut\text{-}off}$ and the difference

in classification accuracy with and without using TDR approach. The difference in classification accuracy is greater than zero when TDR performs better and less than zero when it does not perform better. A particular run is considered acceptable only if the difference in classification accuracy at least equals zero, if not more, with some reduction in the training samples. Hence, difference in classification accuracy is given higher priority followed by the percentage reduction in training samples.

Table 92: Description of markers in graph

Marker	Graph name	Description
⊠	reduced5	Percentage reduction in training samples using 5% pick-up test data
⬒	accdiff5	Difference in classification accuracy using 5% pick-up test data
⊗	reduced10	Percentage reduction in training samples using 10% pick-up test data
◓	accdiff10	Difference in classification accuracy using 10% pick-up test data
⩛	reduced15	Percentage reduction in training samples using 15% pick-up test data
⧋	accdiff15	Difference in classification accuracy using 15% pick-up test data
⨻	reduced20	Percentage reduction in training samples using 20% pick-up test data
◆	accdiff20	Difference in classification accuracy using 20% pick-up test data

Figures 4.6-4.21 show the diagrammatic representation of the performance of TDR approach on the considered eight UCI datasets. Each graph contains a single x-axis along with a double y-axis. The co-ordinates on the graphs are represented by specific marker as described in Table 4.2. Each graph also has a horizontal line at y = 0 corresponding to the secondary y-axis which gives a better visualization as to which is an acceptable result. A result is treated as acceptable if any half-shaded marker is placed on or above this line. The selection with an

acceptable difference in classification accuracy with highest percentage of reduction in training samples is considered as better result.

Figures 4.6-4.13 show the results of using M-αSD based TDR approach for the considered datasets. As shown in Fig. 4.6 for iris dataset, the only selection which placed a half-shaded marker on the horizontal line is at 20% pick-up test data with α=0. But this results in almost 0% reduction in training samples. Hence, no acceptable result is inferred from iris dataset using M-αSD based TDR approach. The reason behind this could be the lesser amount of training samples in the iris dataset and each of them being quite relevant in the classification process. A point to be noted here is that, at α=2 the result is not shown. This is because not a single training sample is selected with this constraint. Again, the result for 5% pick-up test data is also not shown. This is because, a 5% pick-up test dataset is not able to choose at least one test sample from each unique class in the dataset so the percentage of pick-up test data is increased to avoid this. Figure 4.7 shows that the half-shaded markers are placed only at α=0 and 1 for 15% and 20% pick-up dataset for the wine dataset. Among these two, α=1 shows more amount of reduction in training samples which is nearly equal to 60% for both 15% and 20% pick-up test data. For the similar reason as mentioned in case of iris dataset, 5% pick-up test data is not plotted for wine dataset. In case of the glass dataset shown in Fig. 4.8, both 5% and 10% pick-up test data are not able to choose unique samples from the test dataset, hence they are not plotted. This is due to more number of classes in the glass dataset with respect to the total number of test samples. Here, at α=1, 15% pick-up dataset reduces the training samples by nearly 50% giving acceptable results. Figures 4.9 and 4.10 show that for WDBC dataset and balanced scale dataset, the best result is achieved by using 5% of pick-up test data for α=1 which reduces the training samples by nearly 50%. A comparatively higher reduction in training samples i.e. nearly 70% is

achieved by BCW dataset using the same combination of 5% pick-up test data and α=1 shown in Fig. 4.11. Figures 4.12 and 4.13 show that, with α=2 along with 5% pick-up test data, the reduction in training samples for wine quality red and wine quality white dataset is more compared to α=1. The results of these datasets show that with considerable amount of overall samples, 5% pick-up test data produces acceptable results at α=1. Further, it can be inferred that with an increase in the number of overall samples, the α-value can be increased for 5% pick-up test data to produce acceptable results with higher amount of reduction in the training samples using M-αSD based TDR approach.

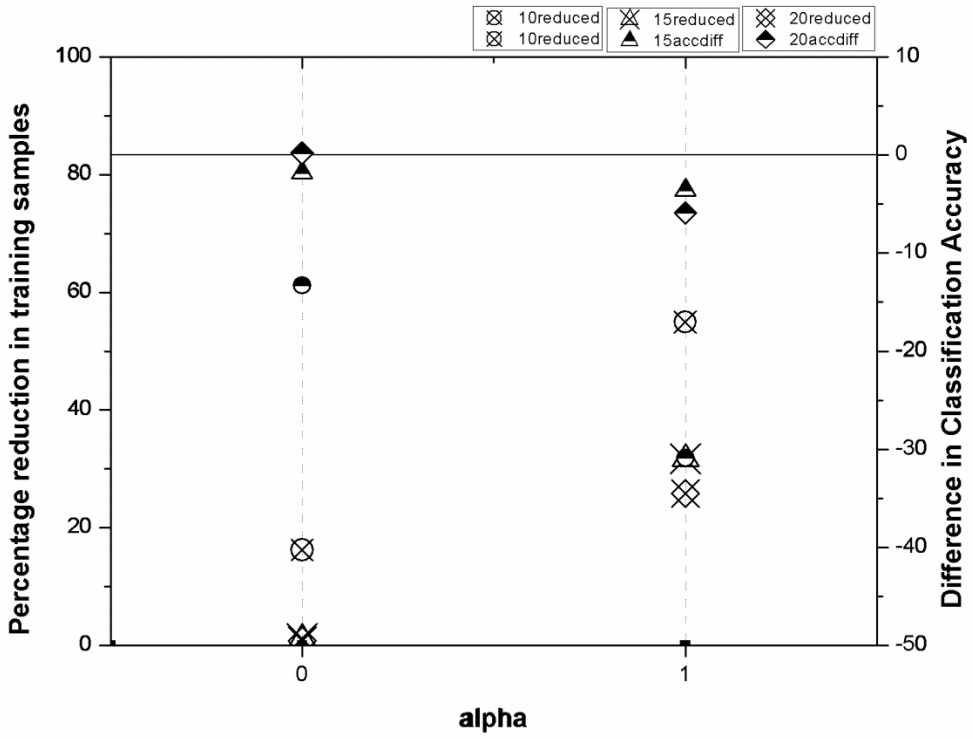


Figure 4.6: Performance of Iris dataset using M-αSD based TDR approach

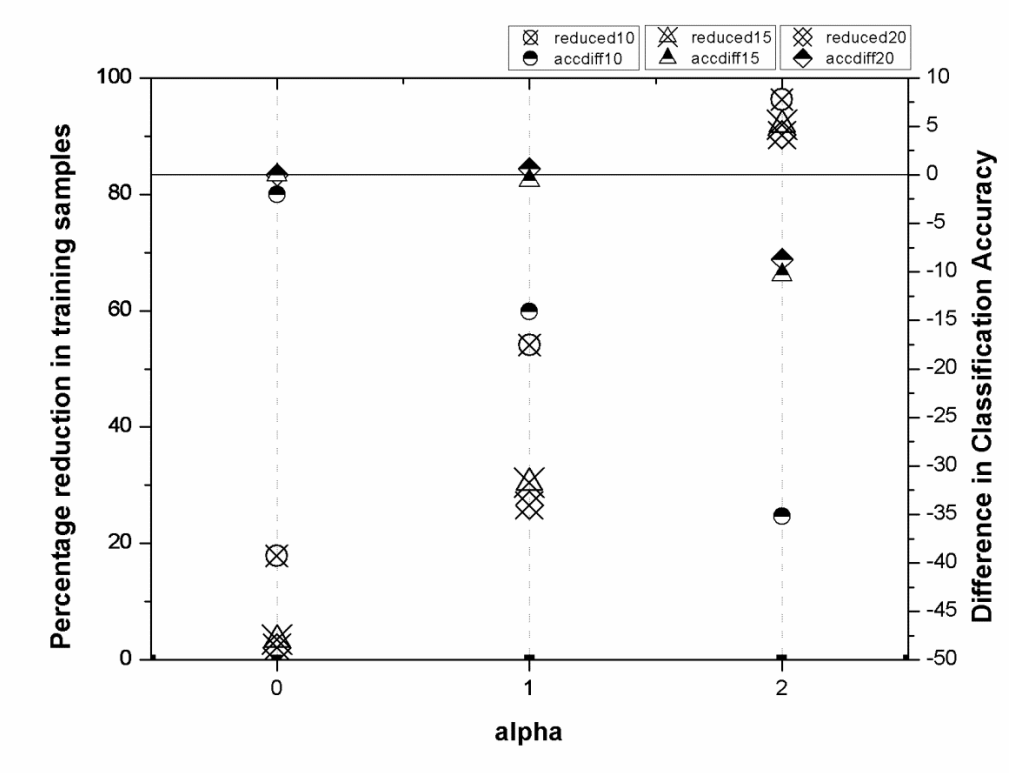


Figure 4.7: Performance of Wine dataset using M-αSD based TDR approach

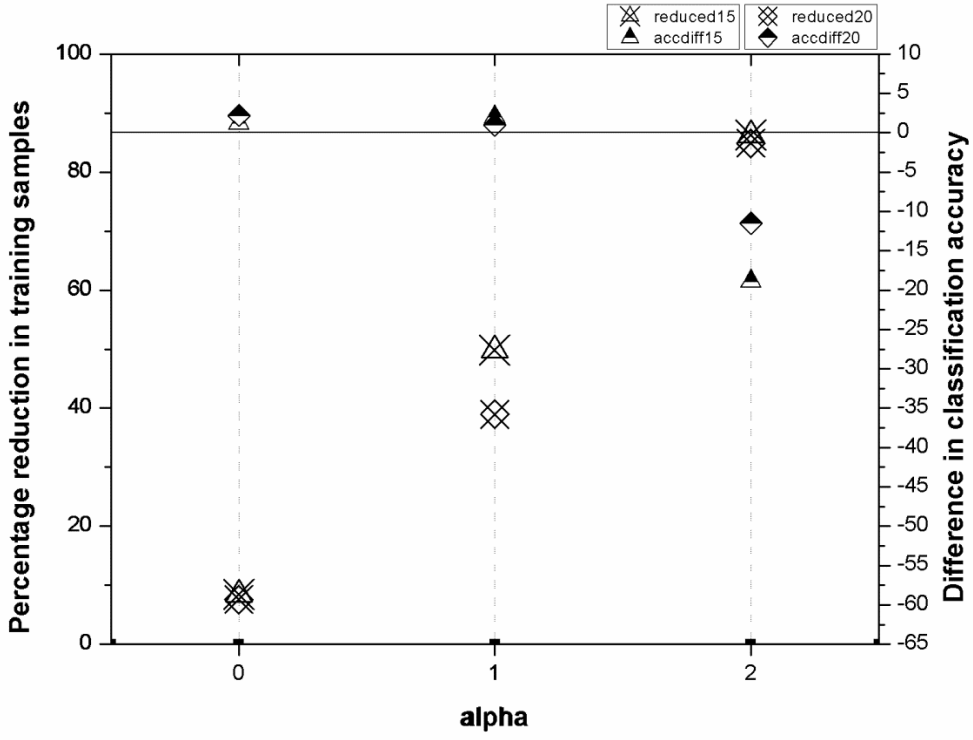


Figure 4.8: Performance of Glass dataset using M-αSD based TDR approach

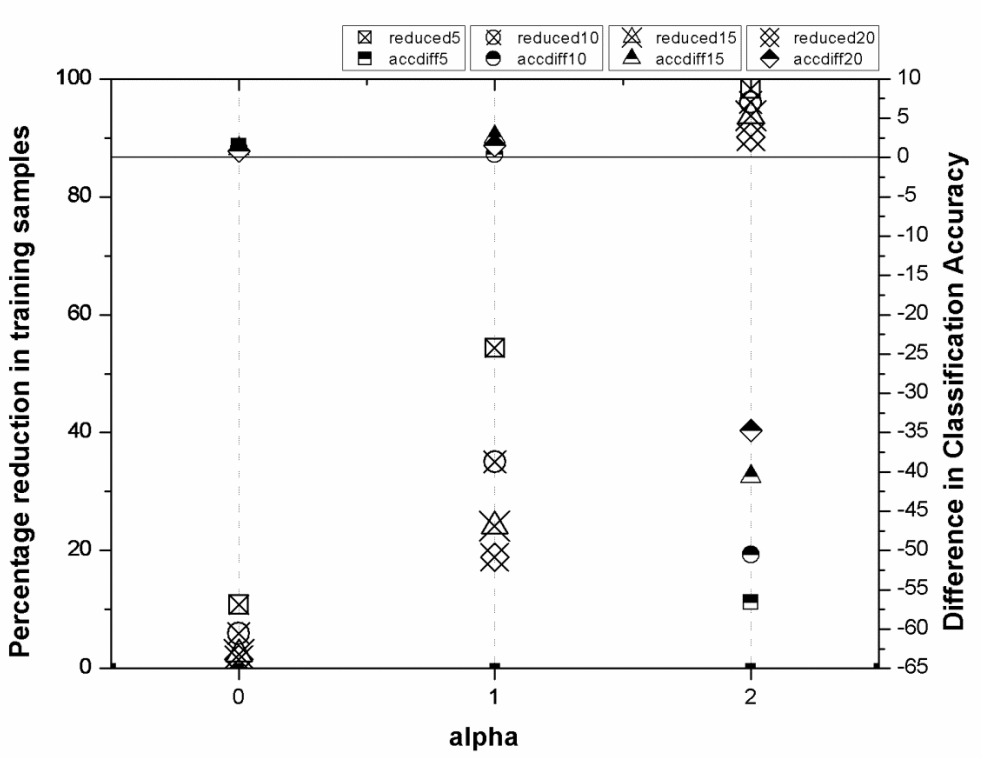


Figure 4.9: Performance of WDBC dataset using M-αSD based TDR approach

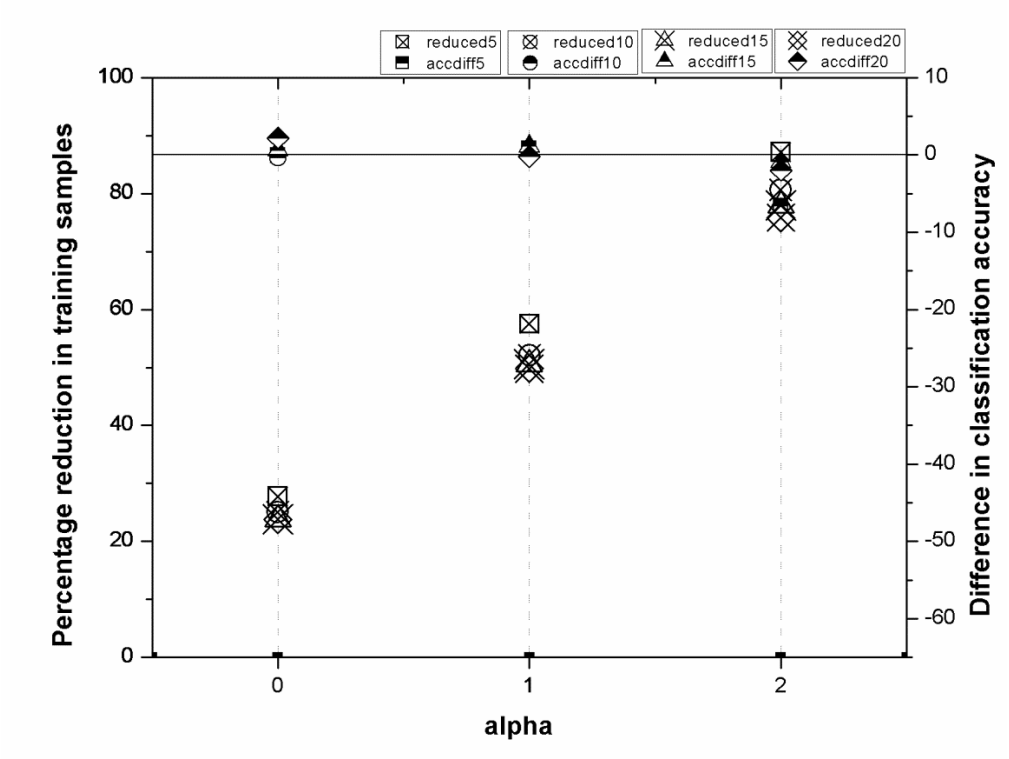


Figure 4.10: Performance of Balanced Scale dataset using M-αSD based TDR approach

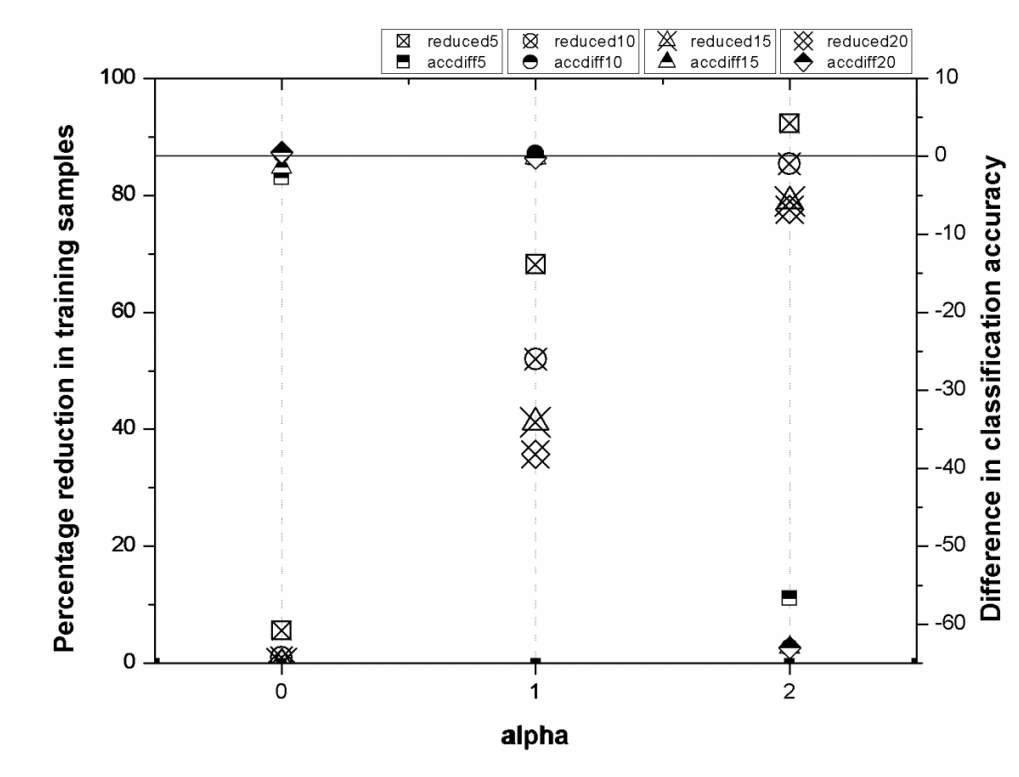


Figure 4.11: Performance of BCW dataset using M-αSD based TDR approach

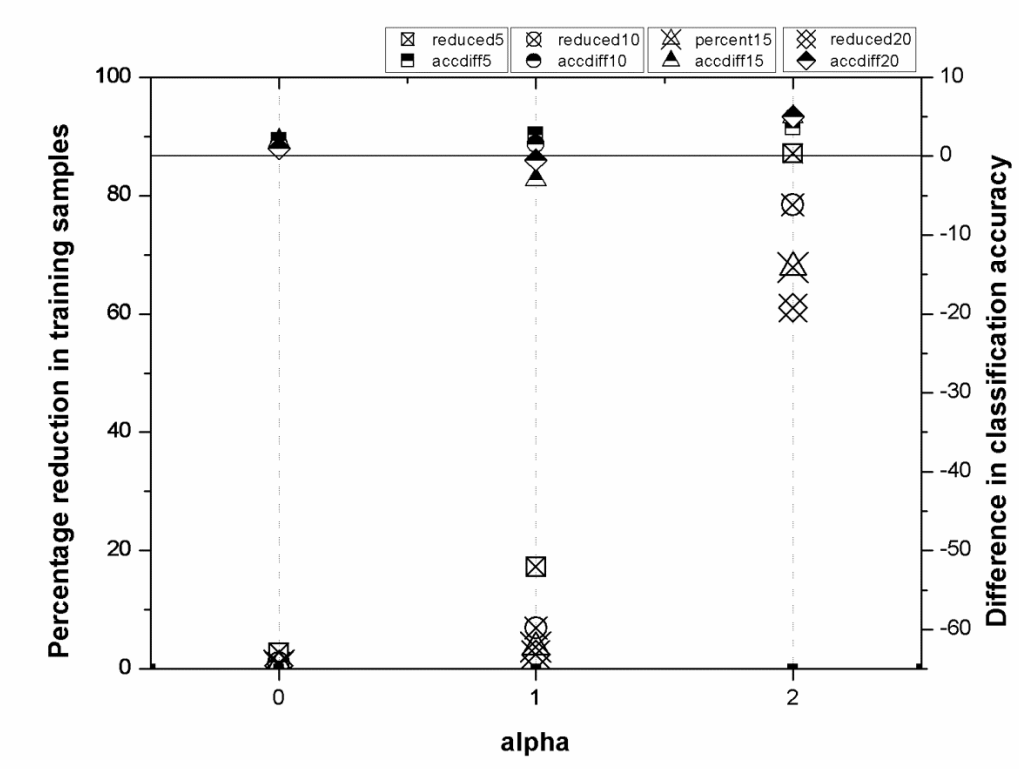


Figure 4.12: Performance of Wine Quality Red dataset using M-αSD based TDR approach

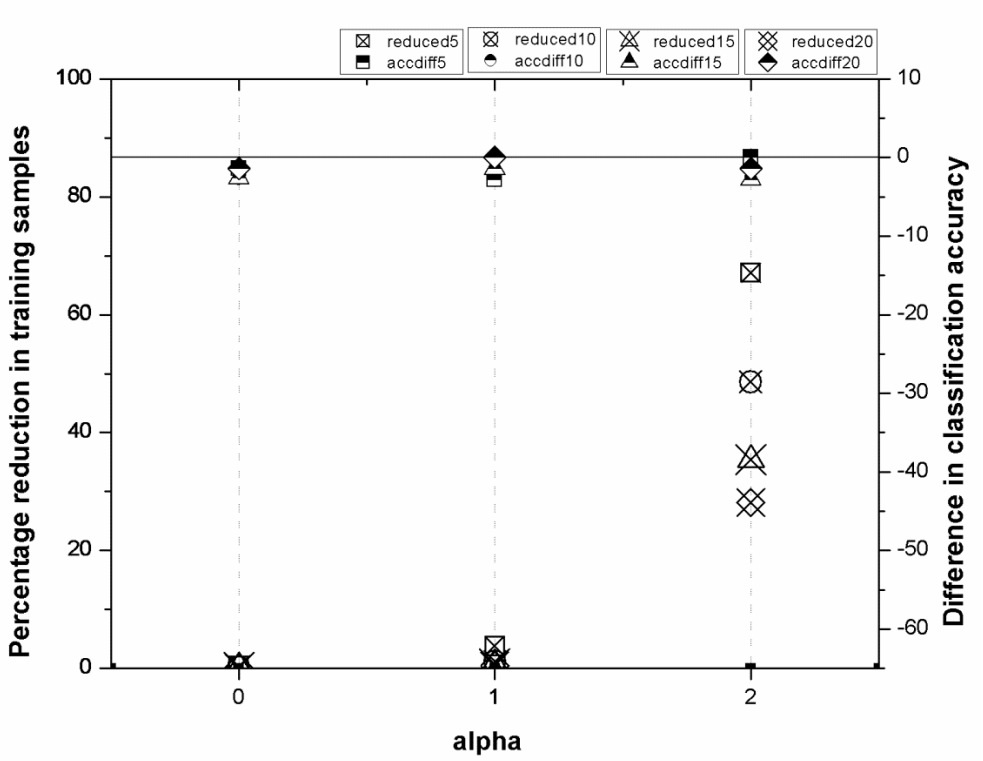


Figure 4.13: Performance of Wine Quality White dataset using M-αSD based TDR approach

Figures 4.14-4.21 shows the results of using area selection based TDR approach for the considered dataset in Table 4.1. As the M-αSD based TDR approach showed acceptable result with 5% pick-up test data, the area selection based TDR approach is also analyzed based on this inference. This approach is adhered in order to cross verify the credibility of M-αSD based TDR approach. Here, for similar reasons as mentioned earlier, the plot for 5% pick-up test data is not shown in Figs. 4.14-4.16 for iris, wine and glass dataset respectively along with 10% pick-up test data for glass dataset. Further, similar to Fig. 4.6, Fig. 4.14 shows that the area selection based TDR approach for iris dataset does not result in acceptable result as all the half shaded markers are considerably below the horizontal line. The result of the wine dataset shown in Fig. 4.15 depicts an acceptable result in classification accuracy at 80-90% area for 10% pick-up test data with a reduction of around 10-20% in training samples. Figure 4.16 shows that for glass dataset, the best result is achieved using 15% pick-up test data using 80% of the area which reduces the

training samples by nearly 20%. These results are because of the lesser amount of samples in these datasets which infers that further reduction in training samples is not necessary. The results of WDBC dataset shown in Fig. 4.17 explains that acceptable results are achieved from 60% area selection with any percentage of pick-up test data. It is obvious that the amount of reduction in the training samples for any percentage of pick-up test data would decrease as the selected area increases. Figure 4.18 shows the same descending trend in the percentage reduction in the training samples for the balanced scale dataset for 5% pick-up test data where 40-60% area produces acceptable results. The acceptable result for the BCW dataset using 5% pick-up test data is achieved at 80% area with a reduction of nearly 25% in the training samples as shown in Fig. 4.19. A drastic percentage reduction in the training samples is seen in the wine quality red dataset as shown in Fig. 4.20. Acceptable results for 5% pick-up test data for this dataset is prominent from 30% area selection with the best reduction of 60% in the training samples. For wine quality white, the acceptable results ranged from 60-80% area for 5% pick-up test data shown in Fig. 4.21.

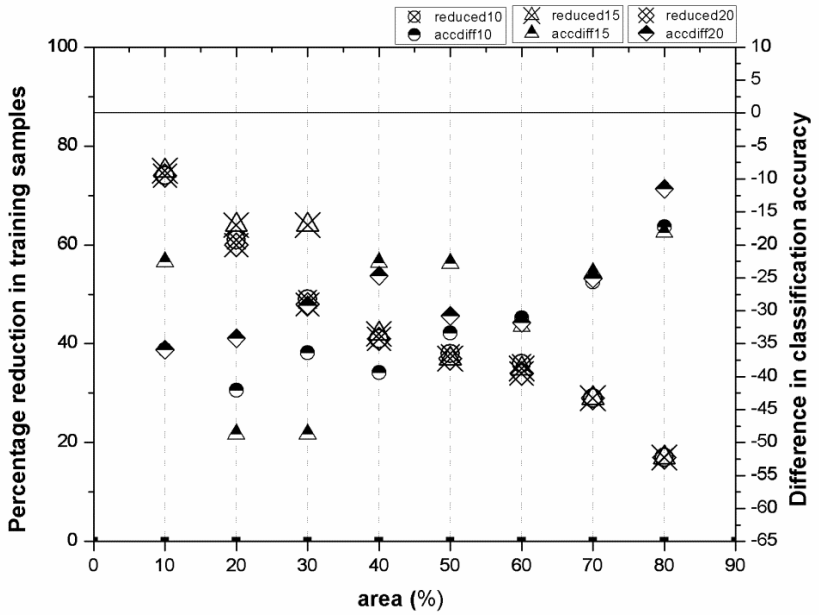


Figure 4.14: Performance of Iris dataset using area selection based TDR approach

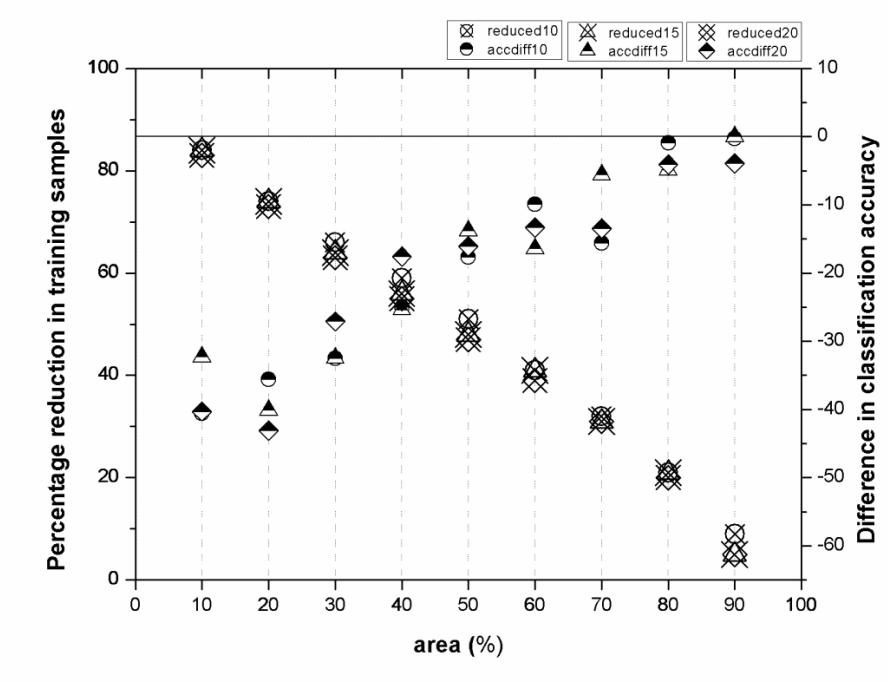


Figure 4.15: Performance of Wine dataset using area selection based TDR approach

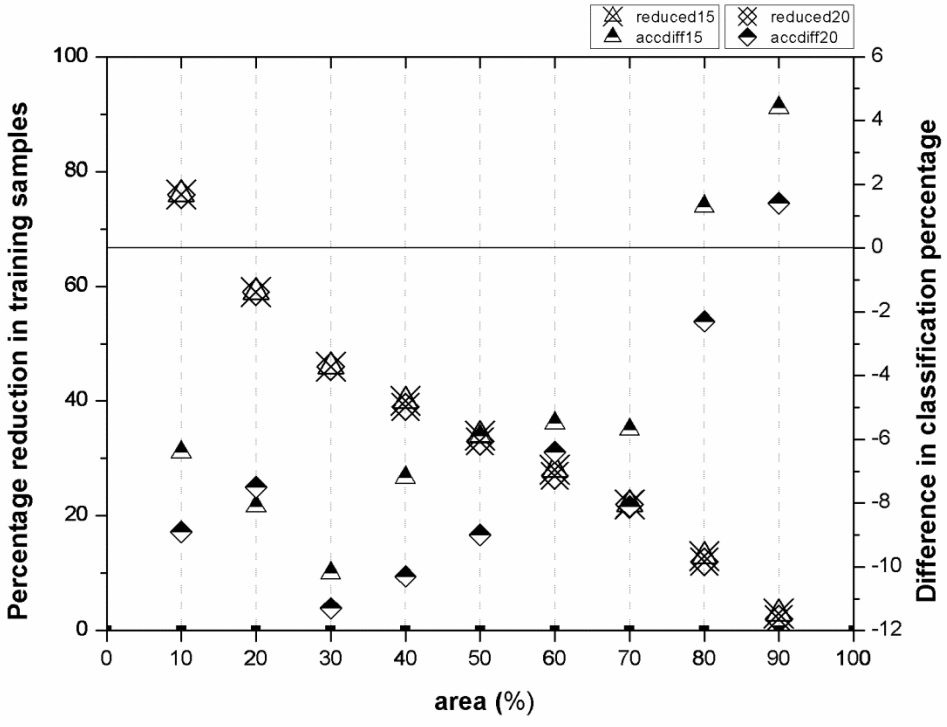


Figure 4.16: Performance of Glass dataset using area selection based TDR approach

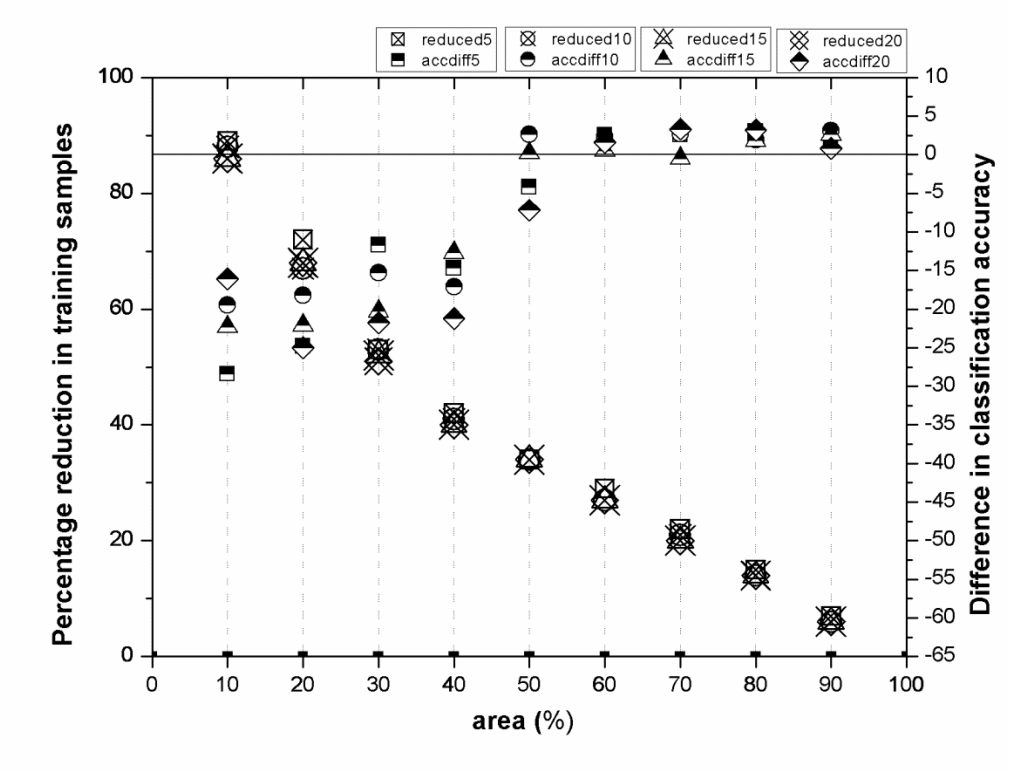


Figure 4.17: Performance of WBDC dataset using area selection based TDR approach

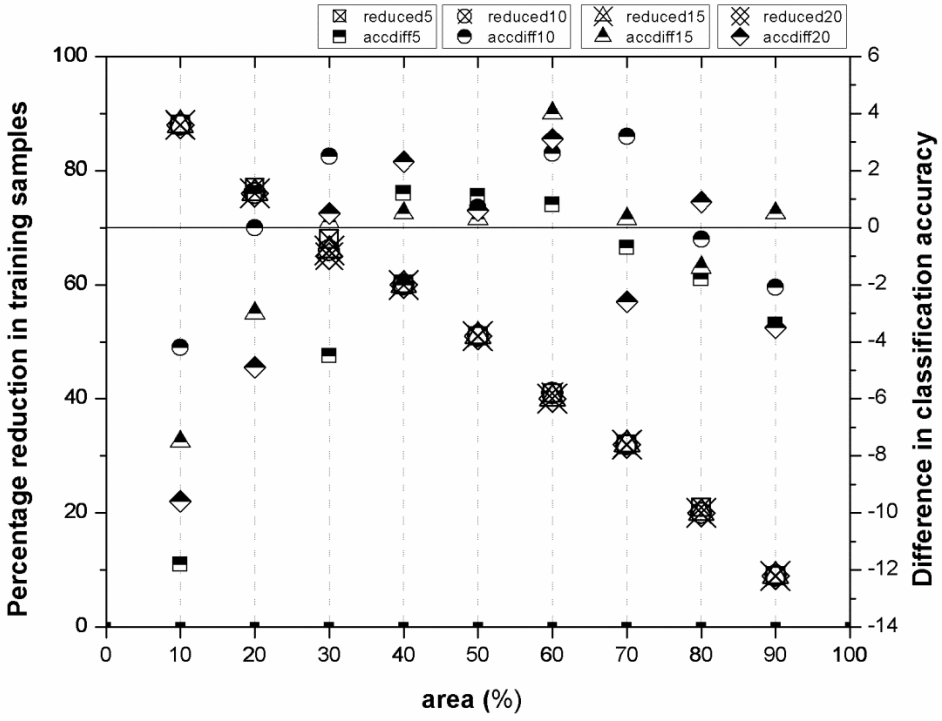


Figure 4.18: Performance of Balanced Scale dataset using area selection based TDR approach

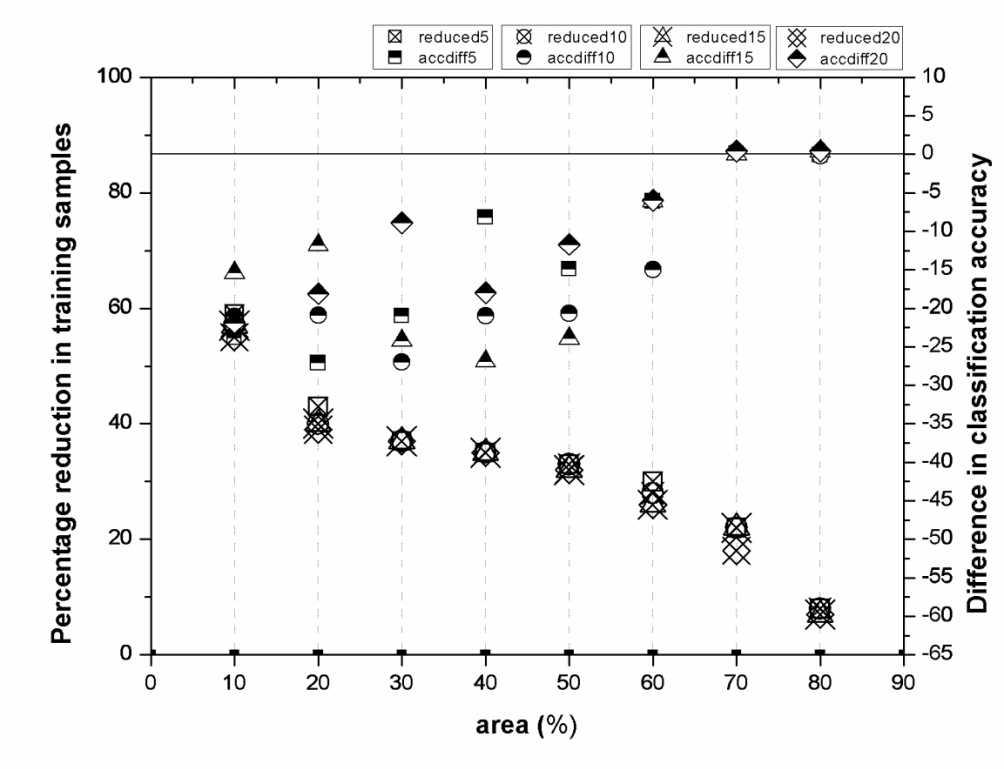


Figure 4.19: Performance of BCW dataset using area selection based TDR approach

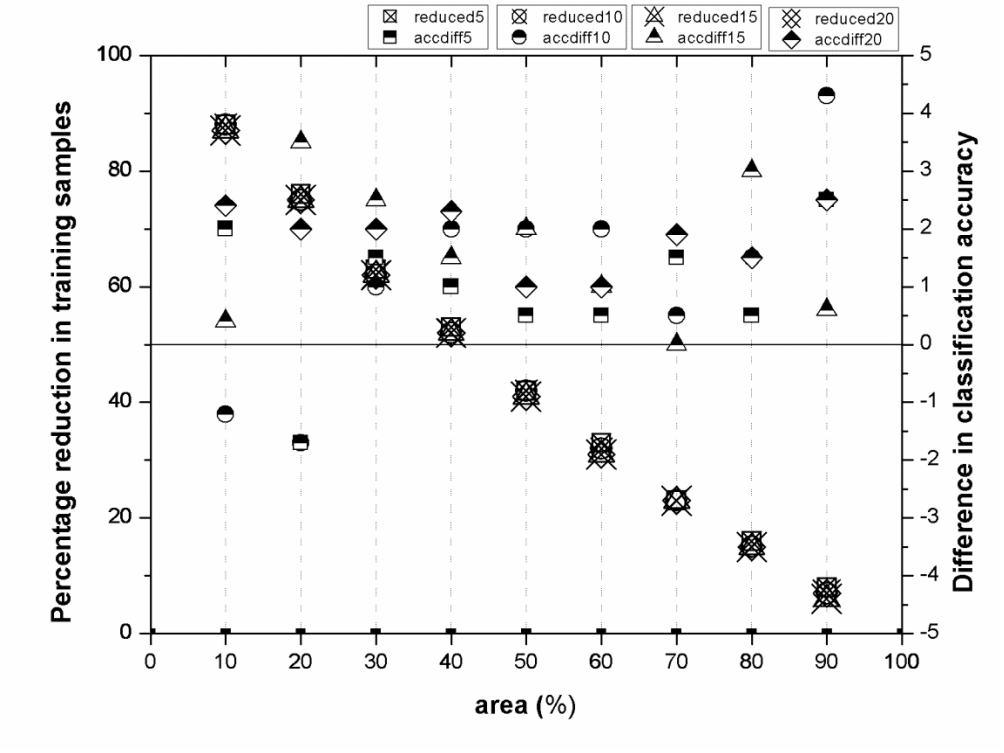


Figure 4.20: Performance of Wine Quality Red dataset using area selection based TDR approach

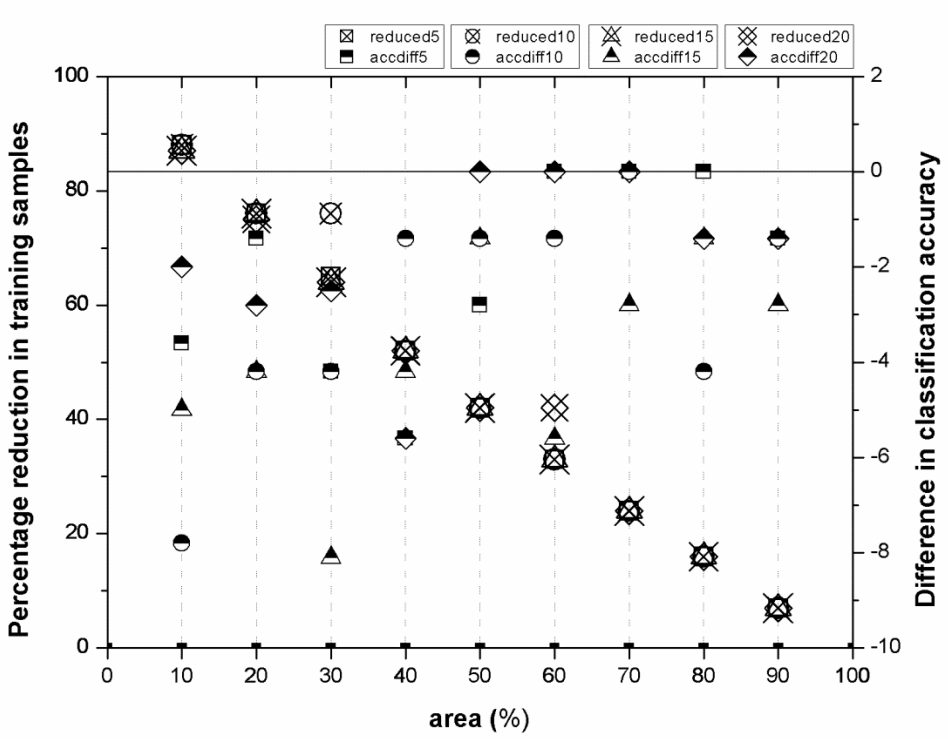


Figure 4.21: Performance of Wine Quality White dataset using area selection based TDR approach

From the above observations, it is inferred that for a considerable amount of samples in a dataset, a 5% pick-up test data is enough to achieve acceptable result using TDR approach. If the number of samples with respect to the number of unique classes is less then increasing the percentage of pick-up test data becomes mandatory. M-αSD based TDR approach uses α=1 for considerable amount of dataset and α=2 for high amount of dataset to get acceptable results using 5% pick-up test data. The area selection based TDR approach gives acceptable results for most of the considered datasets at 60% area using 5% pick-up test data. In the next section, these results are further verified using the transient data of a nuclear power plant.

4.5 Transient classification in a nuclear power plant using TDR approach

As already mentioned, PFBR is a 500MWe pool type sodium cooled fast reactor (SFR) [3] which is in an advanced stage of commissioning at Kalpakkam, India. It consists of main and secondary sodium system along with steam water system as depicted in Fig. 1.1. For demonstration of the TDR method for transient identification in the reactor, the steam water system has been considered. The steam water system consists of steam generators, turbine, condenser, condensate extraction pump (CEP), deaerator, boiler feed pump (BFP) and feed heaters. The two sets of transients that have been considered for investigation are related to two very important pumps in the steam water system known as BFP and CEP. There are three BFP with two running at 50% each and one in stand-by condition. Same is the case for CEP. Each of the two sets of transients consists of two events each making them binary classification problems in each category. The two transients or classes for the BFP category consisted of one BFP trip with stand-by not taking over transient referred as 1-BFP trip and both the BFP trip transient referred as 2-BFP trip. Similarly, in CEP category, the two classes are one CEP trip with stand-by not taking over transient referred as 1-CEP trip and both CEP trip transient referred as 2-CEP trip. The features which are considered to prepare the dataset used the deaerator level (DL) and change in deaerator level (ΔDL) as these two features gives the best result than others as mentioned in Chapter 3. Figure 4.22 and 4.23 shows the time series pattern of DL and ΔDL for the four considered transients. The BFP trip category consisted of a considerable 359 training samples and CEP trip category consisted of as many as 3138 training samples. Resilient backpropagation algorithm is used to train the network which classified the considered transients of PFBR.

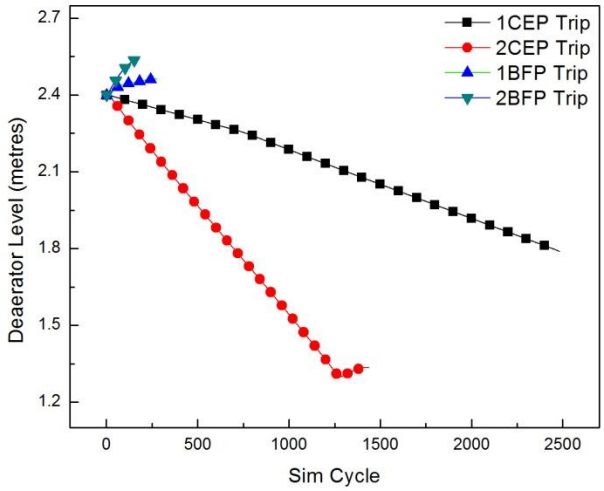


Figure 4.22: Time series pattern of DL for the considered transients

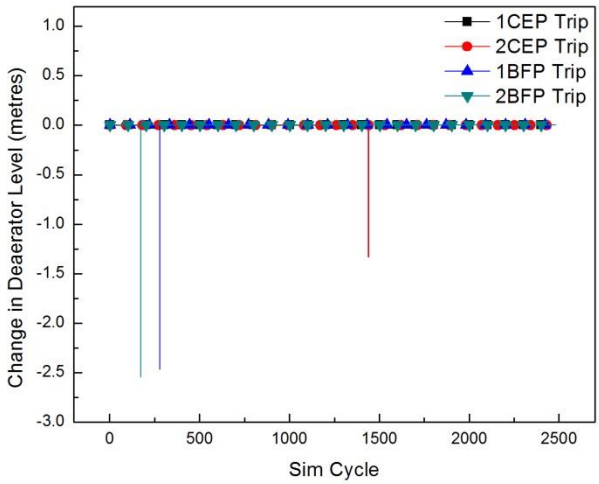


Figure 4.23: Time series pattern of ΔDL for the considered transients

Figures 4.22 and 4.23 show the result of the binary transient classification problem using M-αSD based TDR approach. Both the BFP trips and CEP trips categories are efficiently classified using the TDR approach with reduction in training samples. BFP trip category which has considerable amount of training samples used α=1 to give the acceptable result with nearly 50% reduction in training samples. The CEP trip category which has comparably higher number

of training samples, used α=2 to reduce nearly 70% of the training samples providing acceptable result.

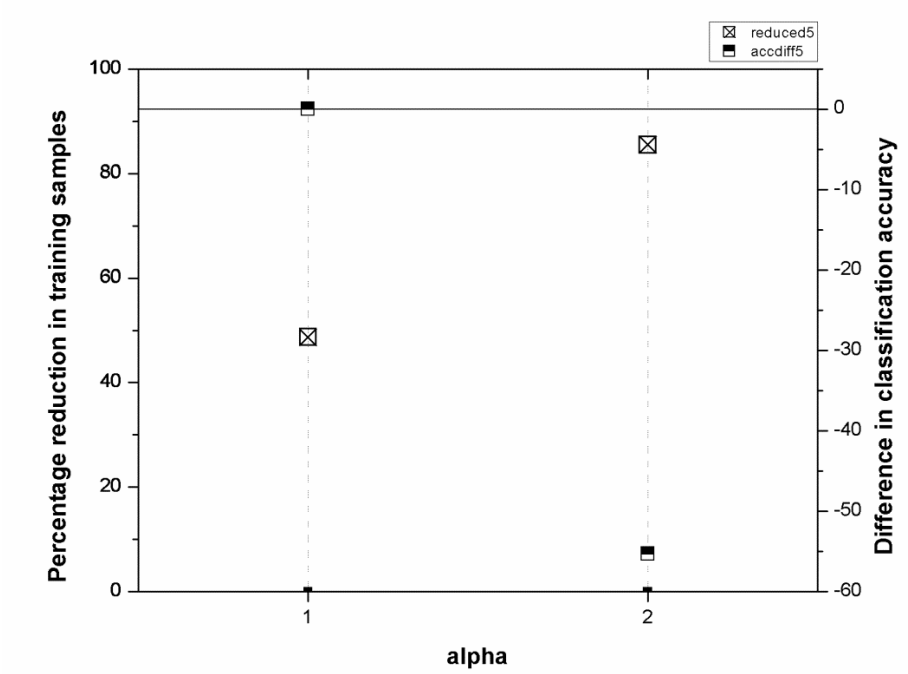


Figure 4.24: Performance of the BFP Trip classifier using M-αSD based TDR approach

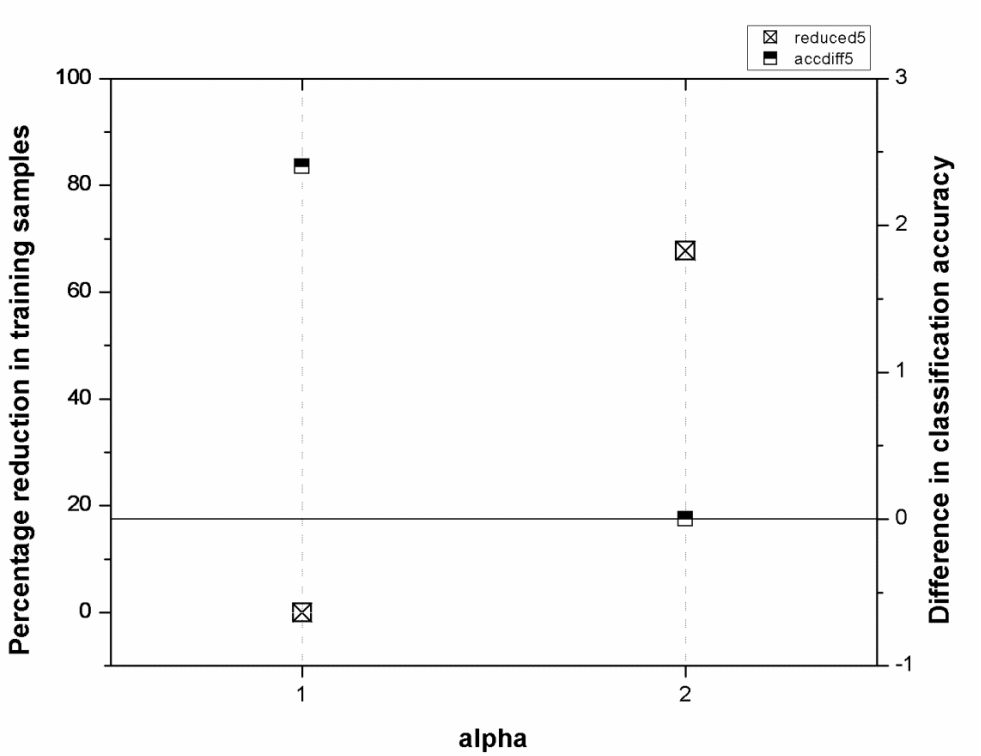


Figure 4.25: Performance of the CEP Trip classifier using M-αSD based TDR approach

Figures 4.24 and 4.25 show the performance of the transient dataset using area selection based TDR approach. At 60% area, the BFP trips reduced the training samples by around 40% with the classification accuracy in the acceptable range. But, the outcome of the CEP trips using 60% area for reduction in training samples resulted in reduction of the classification accuracy by nearly one percent from the previous classification accuracy which is without the use of TDR approach. This shows that, for the considered transient classification problem in PFBR, M-αSD based TDR approach produced better results compared to area selection based TDR approach with a higher percentage reduction in the training samples.

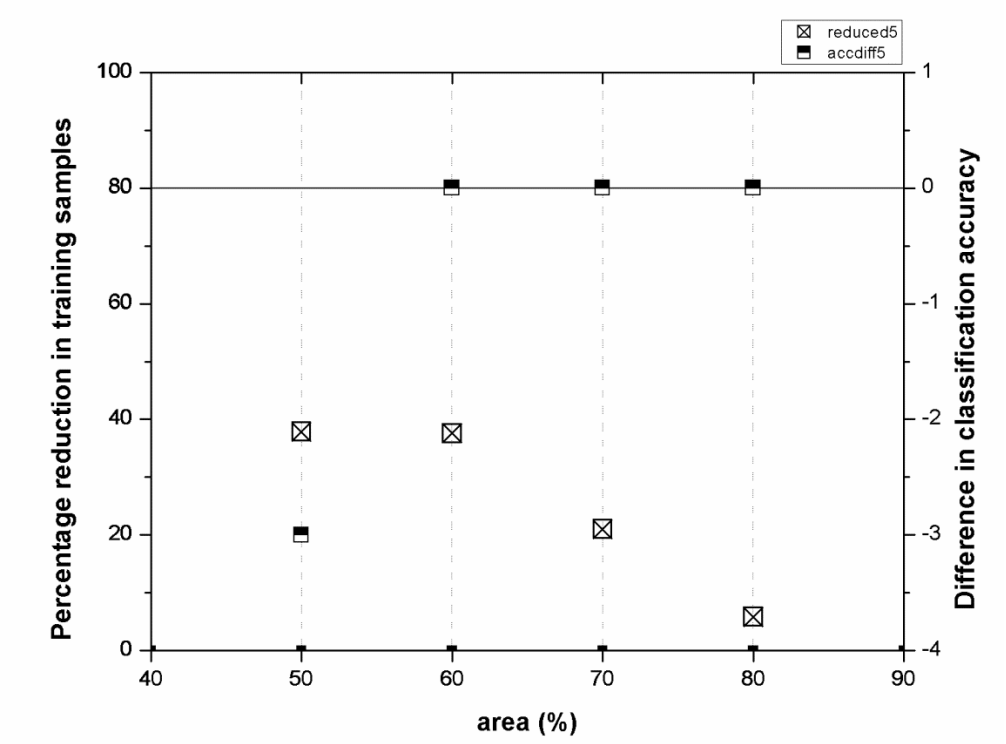


Figure 4.26: Performance of the BFP trip classifier using area selection based TDR approach

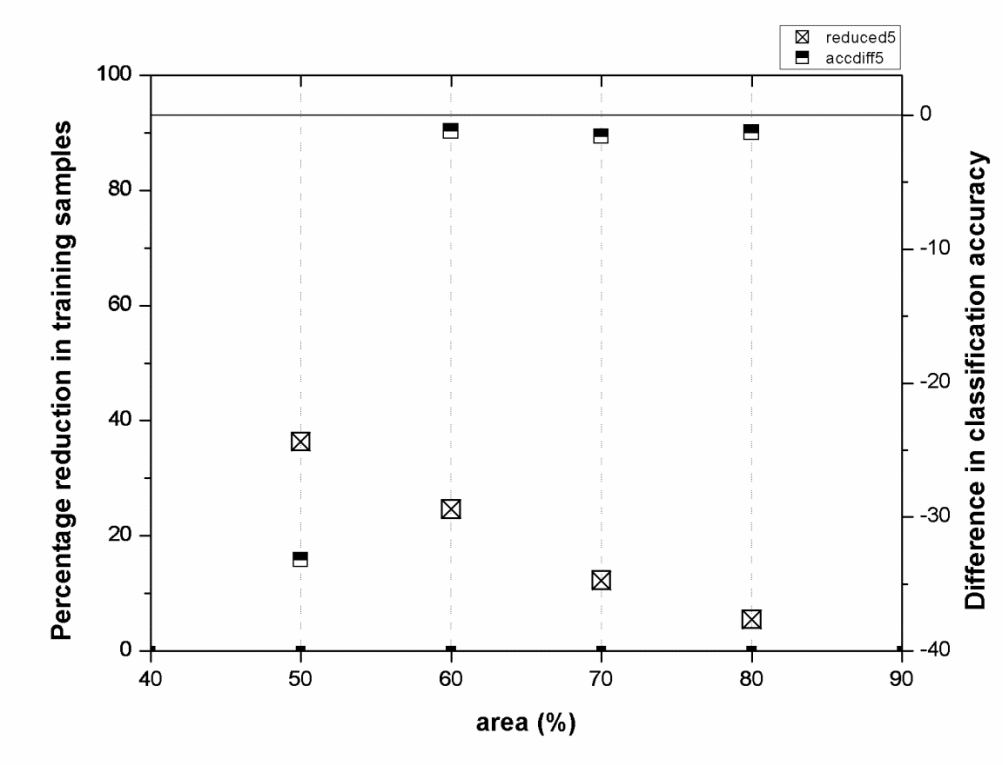


Figure 4.27: Performance of the CEP Trip classifier using area selection based TDR approach

4.6 Summary

The high computational complexity involved in a training algorithm is a serious issue when a massive amount of data has to be dealt with. In order to reduce this complexity for a classifier network, a novel approach such as TDR can be effective.

- This approach reduces the number of training samples which might not be necessary for creation of a hyperspace in order to make a classifier network get trained.
- The Euclidean distance is the distance measure in TDR approach which helps in producing a cut-off boundary for the selection of training samples based on the pick-up test dataset.
- The selected training samples constitute to the new training dataset which is fed as input to the classifier network.

- For the considered real world datasets, it is found that for a 5% pick-up dataset, Mean-αStandard deviation based TDR approach gave acceptable results for α=1 (for considerable amount of data samples) and α=2 (for high amount of data samples) and for area selection based TDR approach gave satisfactory results for 60%.
- For datasets with higher number of classes, the pick-up dataset percentage has to be increased till at least one sample from each class gets selected in the pick-up dataset.
- TDR approach does not produce improved result for the dataset consisting of less training samples compared to the classifier without the implementation of TDR approach.
- It is demonstrated that M-αSD based TDR approach along with the MLP classifier classified the considered transients in a SFR with a better performance with a positive difference in the classification accuracy along with a maximum of about 70% reduction in training samples compared to the non usage of TDR.

As this approach is too juvenile, more refinement to this approach is needed. This refinement might be in the selection of a crisp percentage of pick-up test dataset along with an optimum cut-off Euclidean distance value. An approach to select best of these two features which produce a reduced training dataset that does not compromise on the accuracy of the learning algorithm for the classifier network would make the TDR approach more generalized.

5

DEVELOPMENT OF DIMENSIONALITY REDUCTION BASED CLASSIFIER FOR MALFUNCTION IDENTIFICATION

The present chapter emphasizes on the significance of dimensionality reduction of the dataset for improved performance of a classifier. Principal component analysis, a dimensionality reduction algorithm, is explained in detail along with Adaptive neuro-fuzzy inference system as the classifier. A comparative study is done on the implementation and non implementation of the above mentioned dimensionality reduction algorithm and classifier for classifying events in SFR.

5.1 Introduction

As already explained in previous chapters, a nuclear power plant is a multifaceted engineering composed of many critical yet imperative components. The usage of these components is obligatory. But at times, due to some circumstantial effect, these components are prone to malfunction or may lead to complete failure. These events should be identified well in advance before they lead to any catastrophic results in the plant. In a NPP, during any abnormal incident, the operator stationed in the control room must have the appropriate decision making ability. In order to make such decisions quick, unambiguous and accurate, the overloading of information must be reduced from the operator. Hence, for smooth operation of the NPP, event identification is considered extremely important and a support to the operator [83]. Many soft computing techniques facilitate event identification. As stated earlier, the occurrence of a

transient is designated as an event in NPP. Transient identification in NPP using fuzzy rule based classification system is an example of a soft computing technique being used for event identification delineated in Chapter 2. The neural model is also used for such identification in dynamic processes [84].

An excellent adaptive approach to tune the human-like reasoning capability of a model is well constructed and collated in the adaptive network fuzzy inference system (ANFIS) technique. This technique is used to tune the shape of the membership functions in order to achieve a better result. This is an iterative process where the characteristic features of the membership functions are modified. ANFIS is used in many diversified fields for classification problems. It is used for fault classification in power distribution system [85]. Automatic RNA virus classification using the Entropy-ANFIS method is a novel application of ANFIS [86]. It is also used for fault location in underground cables [87]. Multi-ANFIS or MANFIS is an improvement of ANFIS for multiple outputs [88]

In a NPP, there are almost ten thousand signal data which are sent from the sensors in the plant site to the local control centers. Out of these sensor data, four thousand essential signal data are sent to the main control room which helps to scrutinize the status of the plant. The plant is said to be in normal state or full power when all the sensor signals are within their corresponding threshold limits. During an abnormal state, there is a heavy pandemonium with the alteration in so many sensor signals. During this state, the operators need to have agile notice on many critical signals for taking the best decision to avoid catastrophe. In order to combat this quandary, it is recommended to highlight only the significant and impactful signals. With a large set of sensor data being fed every cycle, an algorithm which discards non critical signals automatically for a particular event could reduce the information overloading on the operator. Hence, principal

component analysis (PCA) can help in processing the sensor data and arrange them in descending order of their significance [89]. Instead of monitoring huge data, PCA helps in discarding the inconsequential data resulting in dimensionality reduction [89, 90]. In dimension reduction process, normalization and standardization are methods used to ensure that the variables receive equal attention [91]. PCA is the basic theory and is widely used to reduce the dimensionality of time series [92, 93]. A novel method is developed for time series data mining known as asynchronous- based PCA [94]. There are instances where PCA have been used for feature extraction using neural network pattern recognition [95].

5.2 Adaptive network based fuzzy inference system

ANFIS is a fuzzy inference system (FIS) using Takagi-Sugeno model which exercises back propagation technique in artificial neural network (ANN). ANFIS is also termed as adaptive neuro-fuzzy inference system. Using this technique, the shapes of the input membership functions are varied in order to reduce the error between the desired output and the actual output of the system. This technique is introduced by Jang [96]. It coalesce the best features of ANN and FIS. In FIS, there is neither a proper procedure to develop the membership function nor the rule base. Hence, it is always a herculean task to modify these two sections if the result is not satisfactory from the FIS. Back propagation algorithm used in ANN comes as a rescuer to this problem. The modification in the shape of the membership functions is done by changing its characteristic parameters. These characteristic parameters are denoted as the weights of the ANN and using back propagation algorithm these weights are modified. Finally, a comparatively reduced error is achieved using ANFIS technique.

Figure 5.1 shows the ANFIS architecture which commonly consists of five layers. For simplicity, only two input variables x_1 and x_2 along with the output variable y have been shown

in the network. Each input variable has only two linguistic values (say less and high). According to Sugeno FIS [97], for a two dimensional input variable containing two membership functions each, there can be four fuzzy rules in the rule set. These fuzzy rules are simple if-then statements which cover the necessary domain of the input variables and can be expressed as

Rule 1 : ***If x_1 is A_1 and x_2 is B_1,***

Then $f_1 = p_1x_1 + q_1x_2 + r_1$.

Rule 2 : ***If x_1 is A_1 and x_2 is B_2,***

Then $f_2 = p_2x_1 + q_2x_2 + r_2$.

Rule 3 : ***If x_1 is A_2 and x_2 is B_1,***

Then $f_3 = p_3x_1 + q_3x_2 + r_3$.

Rule 4 : ***If x_1 is A_2 and x_2 is B_2,***

Then $f_4 = p_4x_1 + q_4x_2 + r_4$.

The consequence functions (f_i) in the rules mentioned above are function of the antecedence variables (x_1, x_2). The coefficients of the consequence function in Sugeno-type FIS (p_i, q_i, r_i) which are otherwise called as consequence parameters are chosen in such a way that it describes the output of the model within the fuzzy region. In ANFIS, the consequence parameters are adaptive in nature so initialization of these parameters at the beginning is done randomly and later on it is modified using least square method during every forward pass of the ANFIS until the final output is achieved.

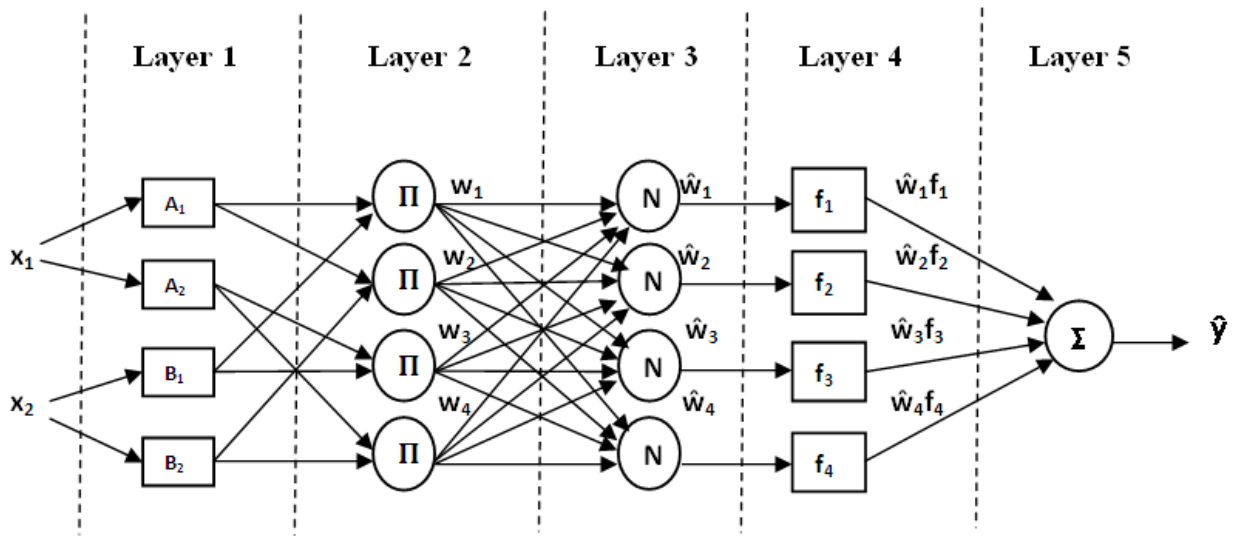


Figure 5.1: Typical architecture of an ANFIS network

5.2.1 Layer 1

Every node in this layer *l* is adaptive in nature. There are two portions as shown in Fig. 5.1. The input variable x_1 along with the linguistic variables, A_1 and A_2 constitute one portion. The other portion x_2 along with B_1 and B_2 is similar to that of the previous portion. The input variables are fuzzified and a membership value 'O_i^1' from each node *i* is obtained.

$$\boldsymbol{O_i^1 = \mu_{A_i}(x_1) \quad ; \quad i = 1, 2} \tag{42}$$

$$\boldsymbol{O_i^1 = \mu_{B_{i-2}}(x_2) \quad ; \quad i = 3, 4} \tag{5.2}$$

Here, x_i denotes the input variable to node *i* and (A_i, B_i) denotes the linguistic variables used to categorize the input variable to node *i*. $\mu_{A_i}(x)$ and $\mu_{B_i}(x)$ are usually bell shaped or Gaussian function which are defined as

$$\boldsymbol{\mu_{A_i}(x_1) = \frac{1}{1 + \left[\left(\frac{x_1 - c_i}{a_i}\right)^2\right]^{b_i}} \quad ; \quad i = 1, 2} \tag{5.3}$$

$$\mu_{B_i}(x_2) = \frac{1}{1 + \left[\left(\frac{x_2 - c_i}{a_i}\right)^2\right]^{b_i}} \quad ; \quad i = 3, 4 \tag{5.4}$$

OR

$$\mu_{A_i}(x_1) = exp\left[-\left(\frac{x_1 - c_i}{a_i}\right)^2\right] \quad ; \quad i = 1, 2 \tag{5.5}$$

$$\mu_{B_{i-2}}(x_2) = exp\left[-\left(\frac{x_2 - c_i}{a_i}\right)^2\right] \quad ; \quad i = 3, 4 \tag{5.6}$$

Here, $\{a_i, b_i, c_i\}$ is the parameter set known as the premise parameters. This is because the shape of the membership function changes based on the parameter set. Hence, the adaptive nature of any node in this layer is typically based on the changes in these parameters.

5.2.2 Layer 2

The circular structure of every node in this layer of Fig. 5.1, labeled **Π**, represents fixed nodes. The output of this node is the product of each and every membership value fed to it and is known as the firing strength of a node.

$$O_i^2 = \mathrm{w_i} = \mu_{A_i}(x_1)\mathrm{x}\ \mu_{B_1}(x_2) \quad ; \quad i = 1, 2 \tag{5.7}$$

$$O_{i+2}^2 = \mathrm{w_{i+2}} = \mu_{A_1}(x_1)\mathrm{x}\ \mu_{B_i}(x_2) \quad ; \quad i = 1, 2 \tag{5.8}$$

5.2.3 Layer 3

Similar to the previous layer, the circular structure of each node in this layer of Fig. 5.1, labeled **N** , represents fixed nodes. This node is used to normalize the firing strengths of each node, so it is named as normalized firing strength.

$$O_i^3 = \hat{w}_i = \frac{w_i}{\sum_{i=1}^{4} w_i} \quad ; \quad i = 1, 2, 3, 4 \tag{5.9}$$

5.2.4 Layer 4

The rectangular structure of each node in this layer of Fig. 5.1 represents adaptive node. Here, each node uses both the input variables along with the normalized firing strength as its node function.

$$O_i^4 = \hat{w}_i f_i = \hat{w}_i (p_i x_1 + q_i x_2 + r_i) \quad ; \quad i = 1, 2, 3, 4 \tag{5.10}$$

Here, f_i is the linear combination of the input variables with $\{p_i, q_i, r_i\}$ as their coefficients. These coefficients are adaptive in nature and are termed as consequence parameters.

5.2.5 Layer 5

The circular structure of the single node in this layer of Fig. 5.1, labeled **Σ,** represents fixed node. The output of this node is the overall output of this network. It results in the sum of all the incoming signals to this node.

$$O_1^5 = \hat{y}_k = \sum_{i=1}^{4} O_i^4 = \sum_{i=1}^{4} \hat{w}_i f_i \tag{5.11}$$

Here, $\hat{y}_k$ denotes the network output for the k^{th} observation. The error in the output at the k^{th} observation is calculated using Eq. 5.12 where y_k is the actual output

$$e = y_k - \hat{y}_k \tag{5.12}$$

The learning of an ANFIS network includes learning of both the premise parameters and the consequence parameters. Hence in such case, the hybrid learning algorithm is used which consists of learning during both forward pass and backward pass [98]. During forward pass, the

consequence parameters in the layer 4 are modified using the least square method whereas during backward pass, the premise parameters in the layer 2 are updated using gradient descent method. Choosing ANFIS over a classifier such as a decision tree is due to some disadvantages related to a decision tree classifier. The main disadvantage of decision tree classifier is its instability. This means that, a minor perturbation in the dataset changes the classifier's performance a lot which refrains it to become a robust classifier which can adjust to noise. This aspect is well taken care of in ANFIS as the error margin and the number of epochs helps to achieve so.

Multiple-ANFIS or MANFIS is used to evaluate network with multiple output where each output is mapped with its independent ANFIS network. This implies that the number of outputs equals the number of ANFIS network. For MANFIS, a nonlinear mapping between the independent variable x and the multiple response y_i, $i = 1,2,\ldots,m$, is achieved by minimizing an error measure E, which is defined as [99]

$$\boldsymbol{E = \sum_{k=1}^{n} \sum_{i=1}^{m} (y_{ik-} \hat{y}_{ik})^2} \qquad \textbf{(5.13)}$$

Here, y_{ik} is the i^{th} actual response for the k^{th} observation, $\hat{y}_{ik}$ is the i^{th} network response for the k^{th} observation and n is the total number of observations.

5.3 Principal Component Analysis

PCA is a technique used to transform a data set to another dimension. This transformation is linear in nature and involves many statistical measures to achieve the transformation. Statistical computational measures such as standard deviation, covariance matrix, eigen values and eigen vectors of the data set are calculated in a procedural manner in order to

get the principal components. These principal components are orthogonal to each other which eventually help in extracting the important information out of the data set. PCA is very popular statistical approach being used for dimensionality reduction [89, 90]. The dimensionality reduction is done taking into consideration that the necessary information in that data set is not lost. One of the areas of practical usage of this technique is for image compression [100–102].

The new orthogonal data set after being linearly projected to another dimension containing a series of principal components. These principal components are arranged in the descending order of the eigen vectors based on the respective eigen values. The eigen vector or the principal component which has the highest eigen value is termed as the first principal component followed by the second principal component and so on till the last column based on the decreasing eigen values. The eigen vectors or the principal components whose respective eigen values are very small are discarded, thereby producing a new matrix *Y*. In such case the initial data set *X* which is of the dimension *r* x *c* is finally transformed and reduced to a matrix *Y* with dimension *r* x *k*, where *k* is less than *c*.

The PCA algorithm is carried out using the following procedure

Step 1. Data preparation

In this step, the data set say X_{rxc} is prepared with *r* rows where each row represents a sample with *c* variables. This data set is also called as data matrix.

Step 2. Mean subtraction

In this step, the mean of each column in the data matrix is calculated and subtracted from every data of their respective columns. Hence, the new data matrix has data with zero mean.

$$\boldsymbol{M_j = \frac{1}{r}\sum_{i=1}^{r} X_{i,j} \quad ; \quad j = 1, 2, \dots, c} \tag{5.14}$$

$$\boldsymbol{X = X - IM} \tag{5.15}$$

Here, I_{rx1} is a unity column vector and M_{1xc} is a row vector with all the mean values of each column [M_j].

Step 3. Covariance Matrix

Covariance is a useful measure which finds out the variance of the data from the mean with respect to other data in that row. The covariance of a data with respect to itself is equivalent to the variance of that data. For example, a 3-dimensional data set (x_1, x_2, x_3), the covariance that can be calculated are *cov*(x_1, x_2), *cov*(x_2, x_3) and *cov*(x_1, x_3). Hence a covariance matrix *C* is produced with *c* x *c* dimension. This is represented as

$$\boldsymbol{C = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) & cov(x_1, x_3) \\ cov(x_2, x_1) & cov(x_2, x_2) & cov(x_2, x_3) \\ cov(x_3, x_1) & cov(x_3, x_2) & cov(x_3, x_3) \end{bmatrix}} \tag{5.16}$$

$$\boldsymbol{cov(x_1, x_2) = cov(x_2, x_1)} \tag{5.17}$$

$$\boldsymbol{cov(x_1, x_2) = \frac{\sum_{i=1}^{r}\{(x_{1,i-\,M_1})(x_{2,i-\,M_2})\}}{(r-1)}} \tag{5.18}$$

Computationally, a much simpler way of doing this calculation is

$$\boldsymbol{C = X^T X \quad ; \quad X = [X_1, X_2, \dots, X_r]^T} \tag{5.19}$$

$$\boldsymbol{X_i = (x_{i,1}, x_{i,2}, \dots\ x_{i,c}) \quad ; \quad i = 1, 2, \dots, r} \tag{5.20}$$

Step 4. Eigen vectors and eigen values

Since the covariance matrix is a square matrix, the calculation of eigen vectors and eigen values are possible. The eigen vectors are arranged based on the descending order of their respective eigen values (λ). The first column in the eigen vector matrix is the one where the data varies the most and is called as the first principal component. All eigen vectors are orthogonal to each other. The second column in the eigen vector matrix is called the second principal component and so on.

Step 5. Selection of basis vectors

This step is important for undergoing dimensionality reduction of the data set as the cut-off factor number or column number eventually selects the basis vectors of the reduced eigen vector matrix. This selection should be done in an optimized manner where dimensionality reduction is achieved without the loss of important information. The basis vectors from all the eigen vectors are selected based on a threshold value α which is dependent on the problem statement. The threshold value is directly proportional to the precision of the classifier output whereas inversely proportional to the amount of dimensionality reduction. This threshold value is compared with the sum of the eigen values (b_k).

$$\boldsymbol{b_k = \sum_{i=1}^{k} \lambda_i} \tag{5.21}$$

If b_k is greater than the threshold value α and k is the cut-off factor number, then the first k eigen vectors are selected as basis vectors. These basis vectors form a new matrix named B_{cxk} where k is less than c. A scree test also helps in deciding the k-value or cut-off factor number [103].

Step 6. Final projected data set

The initial X_{rxc} data set is finally projected on to a new structure with a new set of data matrix Y_{rxk}.

$$\boldsymbol{Y_{r\times k} = X_{r\times c} \times B_{c\times k}} \quad \textbf{(5.22)}$$

This produces a new data set with reduced dimensionality.

5.4 Experimental Procedure

In sodium cooled fast reactors, the cold primary sodium from the cold pool is pumped into the core through two primary sodium pumps (PSP). The cold sodium absorbs the fission heat generated in the core sub assemblies to enter the hot pool. This hot primary sodium then reaches the intermediate heat exchanger (IHX) where it gives away the heat energy to the cold secondary sodium. The cold secondary sodium is pumped into the IHX by two secondary sodium pumps (SSP). The hot secondary sodium finally reaches the shell side of the steam generator (SG). The SG is fed by feed water by two boiler feed pumps (BFP) on the tube side. The feed water evolves as superheated steam at SG outlet by absorbing heat from the secondary sodium. This superheated steam is used to produce electric power in a conventional steam water system. The steam at the turbine exhaust is condensed using condenser which is fed by sea water. The condensate is pumped into the deaerator through two condensate extraction pumps (CEP) which is fed to the SG using two BFPs thus completing the power cycle as shown in Fig. 5.2.

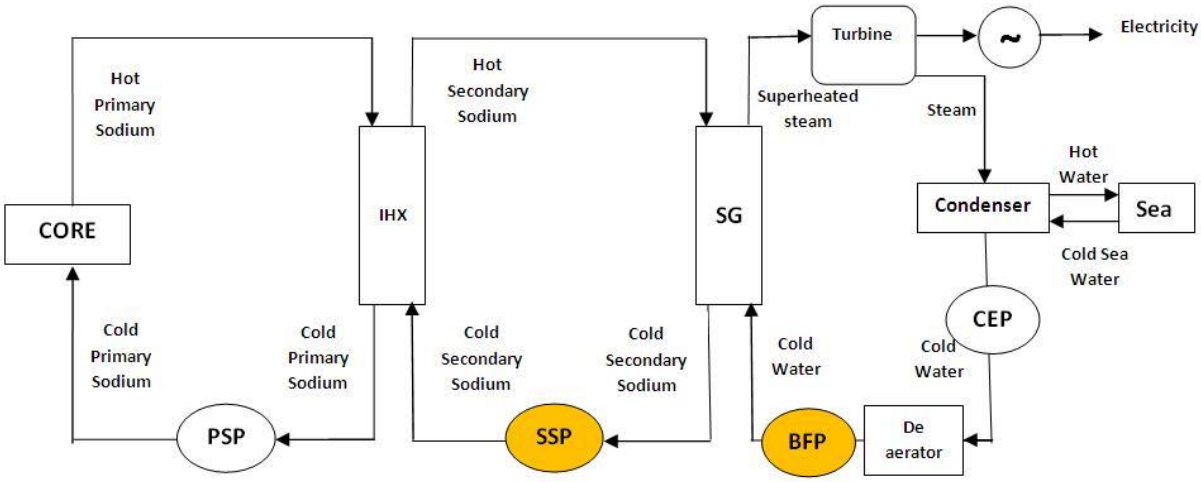


Figure 5.2: Block diagram of the flowchart of PFBR

The operators in the main control room are in charge of the balanced working of the reactor. Along with the critical components, there are components which are comparatively less critical yet important when the component safety is considered. A minute alteration from the normal behavior of characteristic features of these components needs to be noticed, monitored and attended by the operator stationed in the main control room. These small changes can be categorized as some events which in the long run may lead to a bigger mishap if not attended at the right time. For experiment purpose, the events initiated from the components SSP and BFP have been considered for experiment. These two components are shaded in Fig. 5.2. The four primary events which are considered are SSP-1 speed reduction (%), SSP-2 speed reduction (%), feed water flow reduction (%) and feed water temperature reduction (°C). All these events have been simulated on the OTS.

5.4.1 Secondary Sodium Pump (SSP)

The SSP pumps in cold secondary sodium from the SG into the IHX. There are two SSP and four IHX in PFBR (one SSP each for two IHX). The speed of SSP at 100% flow is 900 rpm at an operating temperature of 355°C. In the present experiment, reduction in both SSP-1 speed

and SSP-2 speed has been considered for the analysis. The changes have been simulated for reduction in speed by 10%, 20%, 30% and 40%. Beyond 40% reduction, the reactor shuts down automatically and hence, only these four percentage reductions in SSP-1 and SSP-2 speed have been considered.

5.4.2 Boiler Feed Pump (BFP)

The BFP pumps sub-cooled feed water to SG at 180 bar pressure. This pumping system consists of two 50% turbo-driven BFP and one 50% motor driven BFP which takes over on loss of any of the two turbo-driven BFP. The feed water flow and the feed water temperature are the two characteristic features of the BFP which are considered in this experiment. The operating feed water flow at 100% is 561 kg/s and operating feed water temperature is 235°C at full power. The changes that have been considered and analyzed for feed water flow are reduction in flow by 10%, 20%, . . .,90%. Similarly, the changes in feed water temperature which are considered and analyzed are reduction in temperature by 10°C, 20°C, . . ., 80°C. Here also, temperature reduction beyond 80°C condition is not considered as the reactor goes to shutdown state.

5.4.3 Events and features considered

Table 5.1 shows all the events which are considered for classification. The first bold row denotes the component's parameter that gets affected which can be called as the main or primary event. For example, feed water flow (%) reduced, feed water temperature (°C) reduced, etc. The column to these rows denotes the extent to which the main event has been affected. This can also be called as ancillary or secondary event. For example, by 10, by 20, etc. This shows that there are 25 events in total, i.e., four main events with each having up to a maximum of nine ancillary events. In order to classify all these events, seven common input features are selected from expert's knowledge. These seven features are feed water flow, feed water inlet temperature,

sodium inlet temperature to SG, sodium outlet temperature from SG, SSP-1speed, SSP-2 speed and superheated steam outlet temperature from SG.

Figure 5.3 shows the occurrence of the seven chosen features in the plant schematic in a zoomed in manner. SG being a heat exchanger, the temperature and the flow rate of the liquids entering into both the ends of it are crucial parameters. These parameters are the feed water flow, feed water temperature, sodium inlet temperature to SG and sodium inlet flow to SG. The sodium inlet flow to SG is controlled by SSP-1 and SSP-2. These parameters decide the efficiency of the heat exchanger. The feed water flow is the rate at which the cold water or the feed water from the deaerator is pumped into the SG by BFP. The temperature of this feed water should be maintained above 150°C in order to avoid freezing of sodium during heat transfer. The immediate effect of this heat transfer releases super heated steam at the outlet of the SG with a very high temperature. These seven parameters are the closest to the initiation point of the considered events and get affected immediately. So, these parameters are chosen as the features for the data set. The selection of these seven features for the experiment was based on some prior knowledge of the system where it is assumed that these features contribute in the classification problem. A dimensionality reduction of this dataset is an attempt to verify whether such an effort helps in increasing the performance of the classifier. Hence, though there are only seven features at the outset, an attempt was made to further reducing it. This particular approach can be scaled up where the total number of features is too high and dimensionality reduction would actually end up with only the contributory features.

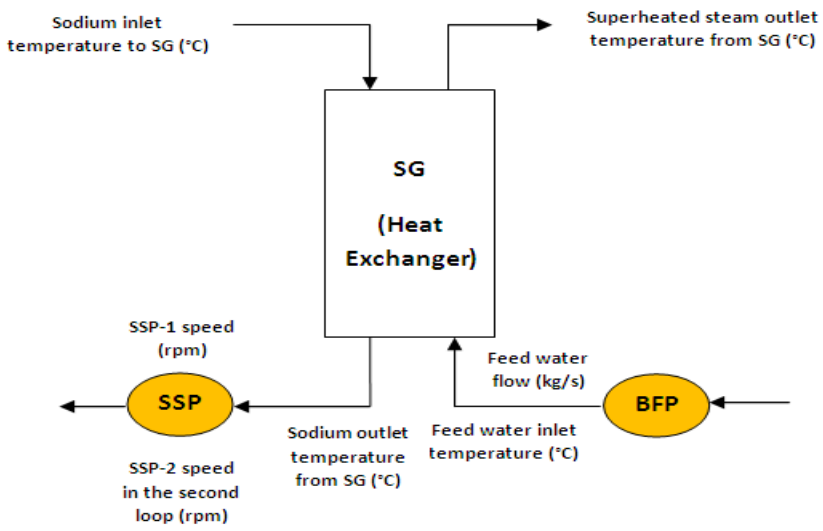


Figure 5.3: Occurrence of the considered seven variables in the plant schematic

Table 10: Events in PFBR considered for classification

Feed Water Flow reduced by	Feed Water inlet temperature reduced by	SSP-1 speed reduced by	SSP-2 speed reduced by
10%	10°C	10%	10%
20%	20°C	20%	20%
30%	30°C	30%	30%
40%	40°C	40%	40%
50%	50°C	-	-
60%	60°C	-	-
70%	70°C	-	-
80%	80°C	-	-
90%	-	-	-

There are two ways of classifying these events. The first approach is similar to the way it looks in Table 5.1. Here, the main event or label-1 event and ancillary event or label-2 event have separate classifiers. The combination of the two outputs from these two classifiers yields the final event. For example, there is an ANFIS-1 which classifies the primary event i.e. say **Feed Water Flow reduced** and the ANFIS-2 classifies the ancillary event i.e. **10%**. These two classifiers combined together give the output as **Feed Water Flow reduced by 10%.** In this

approach, same dataset is fed to both ANFIS-1 and ANFIS-2 during training and testing, and any incorrect output from either of these produces an incorrect classification. In this approach, classifier-1 has to classify 4 classes and classifier-2 has to classify maximum of 9 classes. These variables along with their respective labels form a 'multiple input multiple output' problem for a classifier. The second way being very simple where all these events are classified separately resulting in a single classifier classifying 25 classes. These variables along with their respective label form a 'multiple input single output' problem for a classifier.

5.4.4 Classifier

The classifier chosen in both the cases for experimental purpose is ANFIS. ANFIS is chosen as it has the adaptive nature to adjust the weights of the characteristic features of the membership function in a Takagi-Sugeno fuzzy inference system. This converges to an output with low error. The bell-shaped membership functions have been used for all the input membership functions in this chapter. Each input membership function consisted of a fuzzy set with five fuzzy variables. The five fuzzy variables for each input parameter are named as {VERYLOW, LOW, MEDIUM, HIGH, VERYHIGH}. These five fuzzy variables are able to segregate the input features effectively. Increasing the number of fuzzy variables would increase the complexity of the system and decreasing the number of fuzzy variable would decrease the precision of the output. Hence, five fuzzy variables are chosen as optimum. The problem statement being a multiple output problem in case 1, two ANFIS or MANFIS are used in parallel for the two output labels. Prior to feeding in the input data set to ANFIS for training, a dimensionality reduction of the input data set is performed using PCA as shown in Fig. 5.4. This helps to reduce the complexity of the ANFIS as the number of input variables fed to the model gets reduced. After the MANFIS model got trained, a single row of test data is fed at a time, in

the same way as for training, to check the performance of the ANFIS classifier as shown in Fig. 5.5. It is kept in mind that the number of training data samples (r_1) and test data samples (r_2) need not be equal.

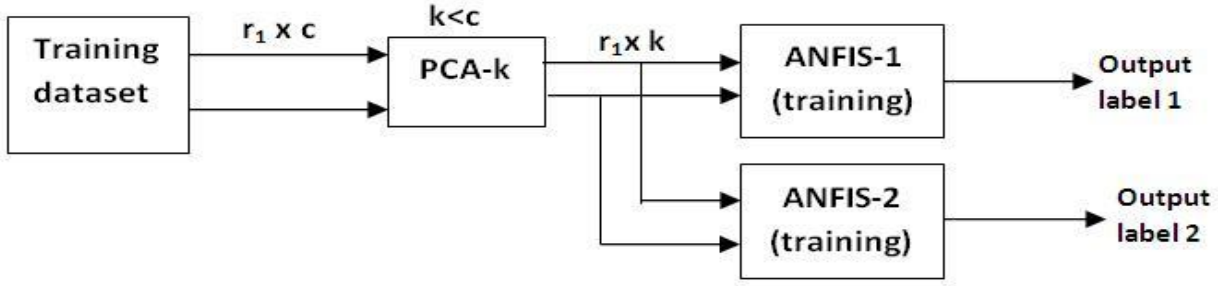


Figure 5.4: Block diagram of the training process in case-1

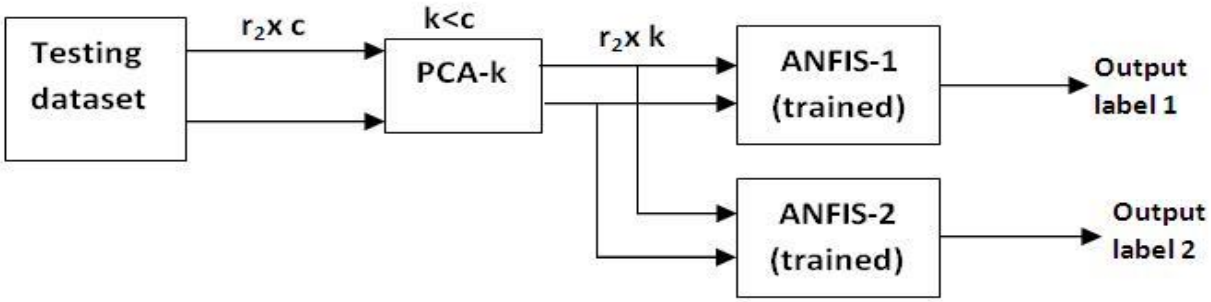


Figure 5.5: Block diagram of the testing process in case-1

5.4.5 Procedure

I. The training data set consisted of r_1 rows and c columns. The rows represent the samples of training data and each row having c columns or features. The output label of each training data row is appended to the training data set.

II. This training data set is fed as input to the PCA which produced a new training data set with (r_1 x k) dimension. Here k is the cut-off factor number and can be any integer less than c.

III. The new training data set is fed to two separate ANFIS models (ANFIS-1 and ANFIS-2), as our problem statement deals with two output labels.

IV. Each ANFIS model gets trained separately producing two independent final ANFIS models.

V. Now, the test data set with dimension r_2 x k is prepared similar to the training data set as in step I.

VI. This test data set is fed to PCA with the same cut-off factor number k as used in the training process.

VII. The new test data set with reduced dimensionality is fed to both the trained ANFIS model which produces two separate output labels known as the ANFIS output event.

VIII. The prediction accuracies of the trained models are calculated by comparing the actual output event and the ANFIS output event.

IX. This process is repeated for various k-values for PCA and different epochs for ANFIS training in order to check at which PCA-k and running the ANFIS for how many epochs, the maximum prediction accuracy is achieved.

This process is made to run for different number of epochs such as 2, 20, 200, 2000 and 20000. As the data set is prepared from the OTS database whose cycle time is 200 milliseconds, a huge amount of data every cycle is generated with very minor changes in their values. Hence, a ten time increment in every epoch level is taken into consideration in order to infer some impactful change in the result.

This process is repeated for a second case where the event data is fed to only one classifier instead of two separate classifiers. Here, the lable-1 events and label-2 events are concatenated as a single event and classification is done.

5.5 Observation

5.5.1 Scree Test

The scree test or scree plot is a plot between the components or factor numbers in the X-axis and the eigen values in a descending order in the Y-axis. The eigen values of the covariance matrix of the training data set is used here. Another way of plotting the scree plot is by taking the percentage of the sum of the eigen values instead of only eigen values. The scree test of the considered training data set is shown in Fig. 5.6. The scree test is a test for determining the number of factors to retain in a factor analysis or principal component analysis [104]. In such a plot, the aim is to look for a 'big gap' or an 'elbow' or a flat line trend. This depicts that the components before this point explains the most about the variability of the data but the rest do not, so the later are discarded [105]. In Fig. 5.6, it is observed that 'big gap' or an 'elbow' is quite evident between factors three and four and a flat trend after four. As the data set is from a NPP which is a safety critical system any loss of information can be disastrous. So, it is advisable to retain the eigen vectors till the fourth principal component instead of third and discard the rest. PCA works on eigen vectors of a covariance matrix. This covariance matrix calculates the variations in each parameter compared to others. The parameters are arranged in descending order of their eigen values which depicts that the parameter which has maximum variance is placed in the first column and the one with the least variance is placed at the last. This transformation gives an insight on those parameters which contribute in the classification and thereby discarding the others which is confirmed by the scree test using the eigen values. This approach eventually helps in dimensionality reduction in a constructive manner by eliminating parameters which show minimal variations or eigen values nearly equal to 0. Hence, the eigen vectors whose eigen values above 0 are selected.

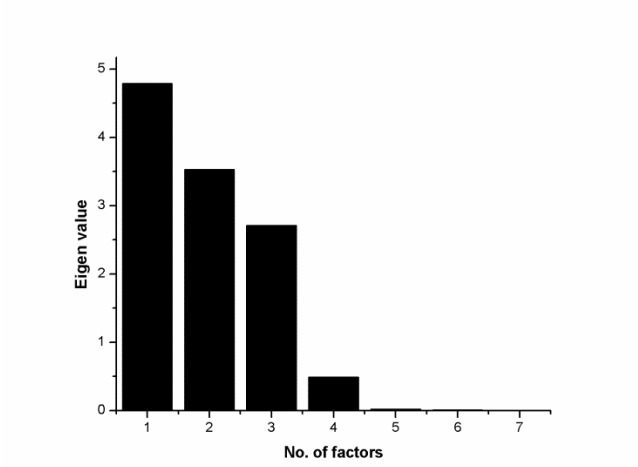


Figure 5.6: Scree test of the training data set

5.5.2 Testing Phase of CASE 1: Using multiple classifiers - MANFIS

In the testing phase, the test data is fed to the trained MANFIS model in order to check its performance. The way of analyzing the performance of a classifier is by plotting a receiver operating characteristics (ROC) as mentioned in Section 1.6 of the present thesis. The ROC curve is the plot between the false positive rate (FPR) or 1-specificity and true positive rate (TPR) or sensitivity in the X-axis and Y-axis respectively. The more the ROC curve tilts towards the left corner of the graph, the better is the classifier. As AUC is the area under a unit square in ROC, its value lies between 0 and 1. The worst classifier which classifies all positive or all negative has an AUC of 0.5. Hence, no realistic classifier should have an AUC below 0.5 [63]. Based on the value of AUC, the quality of the classifier on a general practice is categorized as bad (0.5 to 0.6), fair (0.6 to 0.7), good (0.7 to 0.8), very good (0.8 to 0.9) and excellent (0.9 to 1). An ideal classifier has an AUC value equal to 1.

ROC is used only in the case of binary classifiers. In order to compare the performance of multi class classifiers, a robust classification algorithm based on probability estimation trees is proposed [68]. As the present problem is a multiclass classification problem, this method is

adhered to find the performance of the classifiers. Based on the result from this method, a comparative study on the classifier's performance is done. In this method, for each class, separate ROC curves are plotted taking a particular class as positives and all other classes as negatives. AUC of this curve ($AUC(c_i)$) is calculated and this process is repeated for all the other classes too. The probability estimate of each class is also calculated in a stepwise manner. Initially, difference of each classification output from its actual output is found. The modulus of this value is averaged out among the other classes as this is the error estimate which the network failed to address. The complement of this modulus of error value is the probability estimate of the output being correctly classified. Hence, a vector of probability estimates for the output classes is prepared for each of the test data and a final averaged probability estimate vector is prepared. The final AUC is calculated by summing the product of the AUC of a class with its respective averaged probability estimates ($p(c_i)$).

$$\boldsymbol{AUC_{final} = \sum_{c_i \in C} AUC(c_i) \times p(c_i)} \quad \textbf{(5.23)}$$

Tables 5.2 and 5.3 show the final AUC of ANFIS-1 and ANFIS-2 respectively for different number of epochs (2, 20, 200, 2000 and 20000). In rest of the tables in this chapter, PCA is suffixed with an integer which shows the number of columns chosen from the PCA matrix representing the factor number. PCA1 means only the first column or the first principal component of the PCA matrix is fed as input to ANFIS, PCA2 means the first two columns or the first two principal components are fed as input to ANFIS and so on. PCA0 means the data set is fed directly to ANFIS without undergoing PCA. The histogram representation of the AUC_{final} of both the ANFIS for case-1 shown in Tables 5.2 and 5.3 is depicted in Figures 5.7 and 5.8 respectively.

From the histogram in Fig. 5.7, it is quite clear that the AUC almost remains the same in all cases beyond PCA4 for ANFIS-1. It is also evident that the AUC does not change much from PCA4 to PCA7 for a particular number of epochs. This confirms that no information is lost which could have helped in classifying the events. Moreover, the value of AUC is also in the acceptable range in all these cases, i.e., around 0.99. These values are excellent for any classifier. PCA2 and PCA3 for this case also give good results but as NPP is a safety critical system, PCA4 and above suffice the need. Again, PCA4 and above outperforms ANFIS-1 without any PCA, i.e., PCA0-ANFIS-1. This means ANFIS-1 performance increases when PCA4 and above is used in the preprocessing stage of it. Another analysis which is concluded from Table 5.2 is that there is also not much variation to the AUC values for ANFIS-1 for a particular PCA when the number of epochs is increased. This states that the performances of the classifiers are almost independent of the number of epochs in the training phase. The best AUC for PCA4 is 0.995867 for 200 number of epochs which is almost equal to the overall best AUC, i.e., 0.996197 for 2000 number of epochs for ANFIS-1. According to scree test also, PCA4 is enough to classify the event. Hence, it is wise in considering the former in order to achieve dimensionality reduction.

Table 11: AUC_{final} for test data set in ANFIS-1 of case-1

No. of epochs	PCA0	PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7
2	0.90192	0.759115	0.937026	0.937665	0.98698	0.980936	0.993655	0.993501
20	0.978244	0.759074	0.937396	0.937948	0.986641	0.98487	0.993799	0.992548
200	0.978244	0.759524	0.941398	0.956578	**0.995867**	0.995107	0.993528	0.992922
2000	0.994099	0.789663	0.948224	0.95875	0.995392	**0.996197**	0.984511	0.992529
20000	0.994099	0.789851	0.949562	0.956605	0.994652	0.994846	0.983227	0.992529

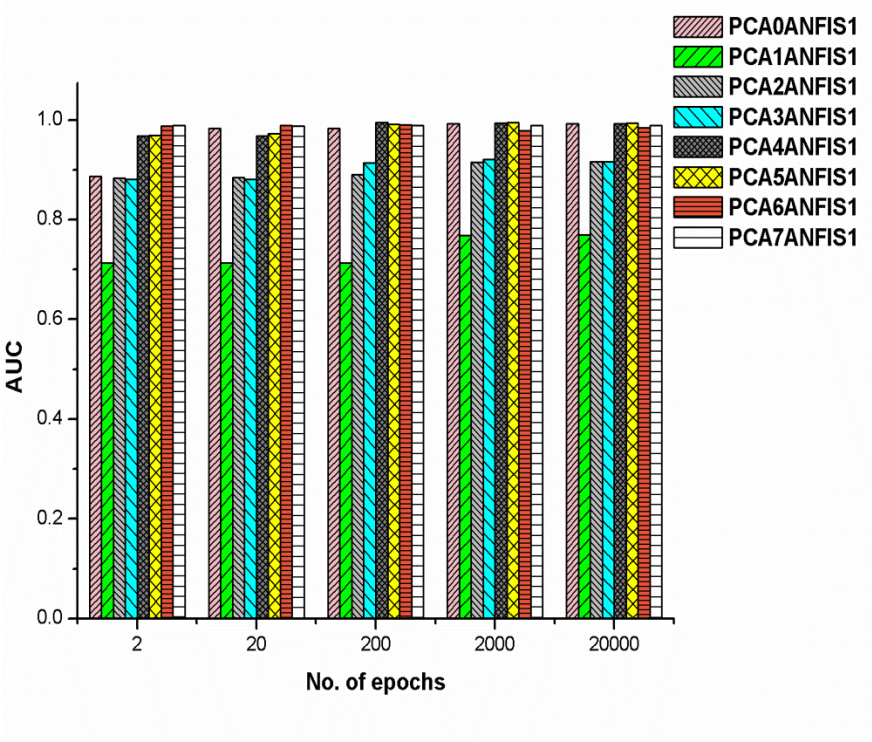


Figure 5.7: Histogram representation of AUC_{final} for test data set in ANFIS-1 of case-1

ANFIS-2 of case-1 classifies the ancillary events or label-2 events and an AUC exceeding 0.8 is in the acceptable range as these events are secondary events. In Table 5.3, it is clear that the AUC of PCA4 and above for ANFIS-2 is greater than 0.85 with the best AUC of 0.903173 from PCA7 running for 2 epochs. Thus, this range of PCA for ANFIS-2 can be considered as very good category based on their AUC values. Figure 5.8 shows the histogram of ANFIS-2 which shows the final AUC is almost the same for factor number 4 and above for different number of epochs. Similar to ANFIS-1, the optimized cut-off factor number to be considered for ANFIS-2 considering dimensionality reduction is PCA4 for 2000 number of epochs. Finally, from the AUC analysis, it is concluded that PCA4 for both ANFIS-1 and ANFIS-2 for case 1 gives the acceptable results. This conveys a reduction in the dimensionality of the data set which is fed as input data to the classifiers.

Table 12: AUC_{final} for test data set in ANFIS-2 of case-1

No. of epochs	PCA0	PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7
2	0.76471	0.573468	0.503198	0.620132	0.8527	0.855959	0.881403	**0.903173**
20	0.767857	0.573858	0.507918	0.624737	0.859362	0.856954	0.884051	0.90195
200	0.838264	0.611275	0.612924	0.717984	0.851253	0.868476	0.876478	0.903064
2000	0.838264	0.56137	0.673773	0.661573	0.892327	0.882119	0.884443	0.901852
20000	0.838264	0.56137	0.674448	0.649011	0.890361	0.891512	0.898945	0.901852

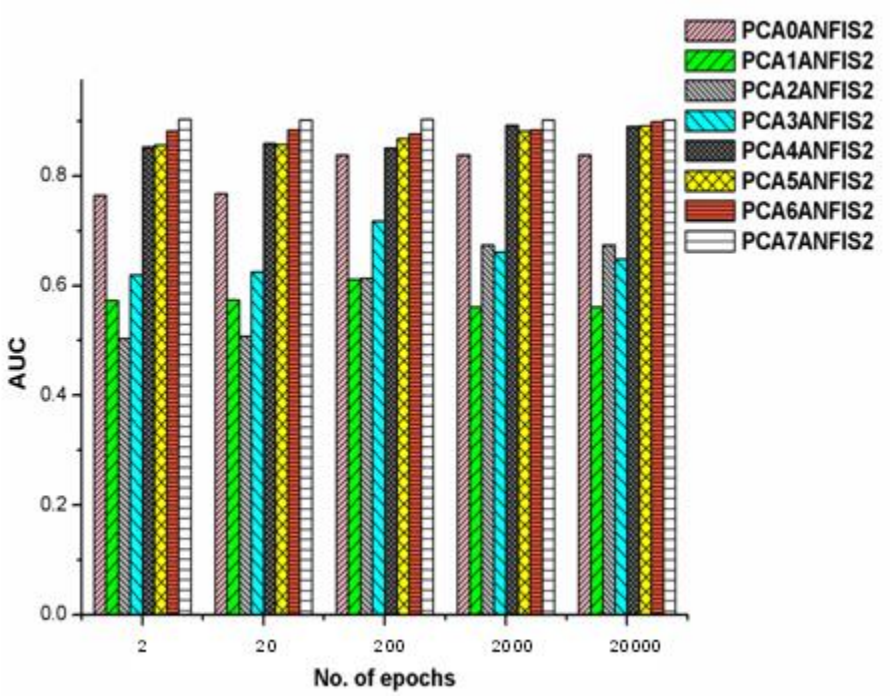


Figure 5.8: Histogram representation of AUC_{final} for test data set in ANFIS-2 of case-1

5.5.3 Test Phase of CASE 2: Using one classifier

Similar to case 1, the AUC of the classifier is calculated and shown in Table 5.4. It is evident from Fig 5.9 that the AUC values for the classifier for cut-off factor number 4 and above produces better results as compared to others. This matches with the result from the scree test. But the AUC values ranges from around 0.7 to 0.9. The best AUC is found to be 0.900634 in

PCA5 for 20000 number of epochs. These results are better than the ANFIS performance without PCA for the considered number of epochs shown in the column PCA0. Figure 5.9 shows the histogram representation of Table 5.4.

Table 13: AUC_{final} for test data set in ANFIS of case-2

No. of epochs	PCA0	PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7
2	0.693679	0.501235	0.50575	0.564382	0.756205	0.720804	0.712573	0.760396
20	0.698917	0.501429	0.506841	0.564099	0.755822	0.716565	0.707971	0.752115
200	0.692521	0.500572	0.54549	0.498386	0.768584	0.752258	0.707247	0.743769
2000	0.726109	0.604956	0.593397	0.547559	0.743646	0.850903	0.81728	0.743769
20000	0.725871	0.58395	0.592959	0.572385	0.774226	**0.900634**	0.856322	0.743769

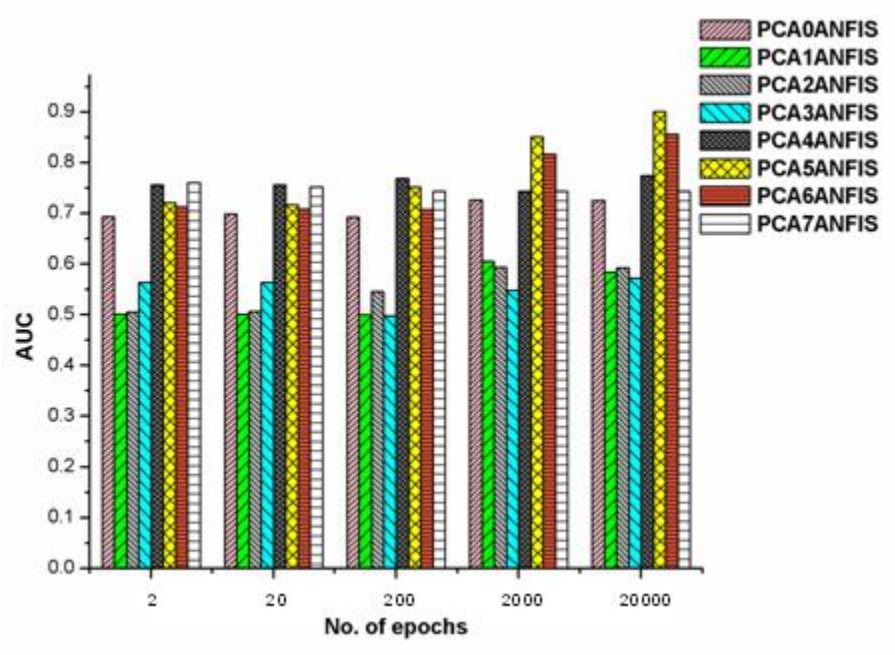


Figure 5.9: Histogram representation of AUC_{final} for test data set in ANFIS of case-2

5.6 Results

Logically comparing the values of AUC for different PCA in case-1 and case-2, PCA4 in case-1 performs optimally better among all the others. It is necessary now to choose the optimal

number of epochs that both ANFIS-1 and ANFIS-2 should iterate in order to get the best result. This can be found using Table 5.5 which appends the values of AUC for PCA4 only, from Table 5.3 and 5.4. As mentioned earlier, label-1 events classified by ANFIS-1 is the primary event, so it has more priority and should have the best AUC among all. Table 5.5 shows that, for 200 epochs, the AUC for ANFIS-1 is best compared to all others in its column. Even though its corresponding AUC value of ANFIS-2 is not the best in its column, iterating for 200 epochs for our experiment is wise as our priority is label-1 events. Hence, logically for our experiment, in order to classify a multiple output event, it is wise to iterate MANFIS for 200 epochs using PCA4 for dimensionality reduction. The ROC curves for the best performers, i.e., PCA4 for ANFIS-1 and ANFIS-2 of case-1 for 200 epochs are shown in Figs. 5.10 and 5.11.

Table 14: Comparison of AUC_{final} of PCA4 from Tables 5.3 and 5.4

No. of epochs	**ANFIS-1**	**ANFIS-2**
2	0.98698	0.8527
20	0.986641	0.859362
200	**0.995867**	0.851253
2000	0.995392	**0.892327**
20000	0.994652	0.890361

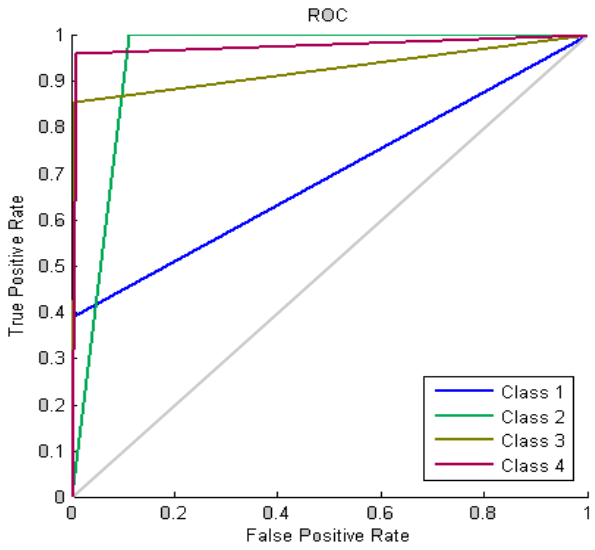


Figure 5.10: ROC curve of PCA4 in ANFIS-1 of case-1 for 200 epochs

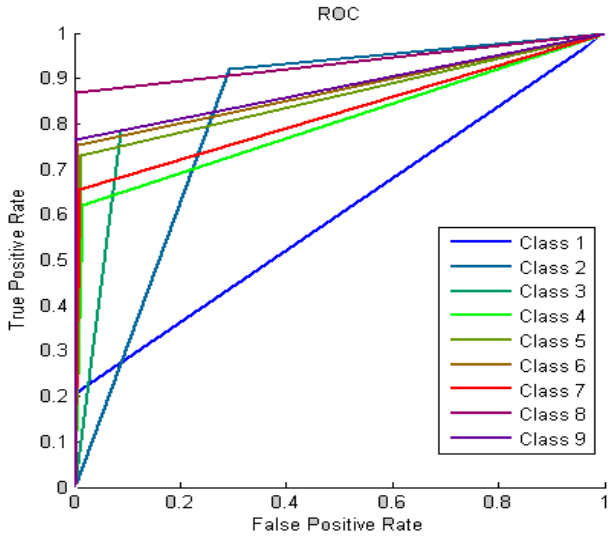


Figure 5.11: ROC curve of PCA4 in ANFIS-2 of case-1 for 200 epochs

Using single ANFIS in case-2, it is observed that the largest AUC of 0.900634 is by PCA5 based ANFIS classifier at 20000 number of epochs. Comparing, the AUC values of MANFIS with two classifiers and ANFIS with one classifier, it is observed that ANFIS-1 of MANFIS which classifies the primary event produces far better AUC than single ANFIS which classifies a combined event. It has been also stated earlier that the primary event is given highest

priority compared to others. The difference in these two AUC values is too big considering any classification problem. Hence, PCA4 based multiple ANFIS classifier at 200 number of epochs produces better performance than PCA5 based single ANFIS classifier at 20000 number of epochs. In simple terms, the best among the multiple ANFIS gives better performance than the best among the single ANFIS classifier using PCA as dimensionality reduction.

5.7 Summary

This chapter outlines the feasibility check on the practice of dimensionality reduction using PCA for event classification in a SFR which is a complex critical system where safety is of utmost priority. The significant inferences that can be extracted from this chapter are as follows.

- PCA along with ANFIS can be used as a dimensionality reduction based classifier for event classification.
- It performs better than an ANFIS classifier without dimensionality reduction for the considered events.
- Scree plot is a measure to find the cut-off factor number for dimensionality reduction.
- PCA based MANFIS produced better performance with higher AUC compared to PCA based ANFIS.
- Hence, dividing an event to sub events like label-1 event or primary event and label-2 event or ancillary event and then classifying it using MANFIS classifiers gives better performance measures compared to a single ANFIS classifier.

6

DEVELOPMENT OF WEIGHTED SMOTE ALGORITHM FOR IMBALANCED DATASET

The present chapter enlightens on the imbalanced dataset problem and emphasizes on the oversampling method as solution to this issue. Synthetic minority oversampling technique which is a very common approach to address the imbalanced dataset issue has been the focus of the study. A modification to this algorithm using weighted approach has been elucidated along with the comparison between both using some real world datasets. The feasibility of this weighted approach for identifying some malfunctions in SFR has also been studied in this chapter.

6.1 Introduction

The necessity of improved performance of a classifier network in classifying minority data samples in an imbalanced dataset has brought in a lot of awareness amongst researchers and users. An imbalanced dataset with two-classes consists of data samples with a huge difference between the number of minority data samples and the majority data samples. The minority dataset consists of the samples of a particular class those are less in numbers whereas the majority dataset consists of the samples of the other class those are comparatively more in numbers. Such kind of imbalance in dataset is called as between-class imbalance [106] compared to within-class imbalance [107]. The performance of the classifier network for such imbalanced dataset is always biased towards the majority dataset because of more number of samples it contains. Hence, the classifier does not classify the minority data samples accurately and more

often than not these samples are misclassified. This leads to a big trouble in cases where classifying the minority data samples are of utmost priority compared to majority. An example of such scenario is classifying the occurrence of a malignant disease among a group of people who have symptoms of that disease. In such a case, very few people will actually have a malignant disease compared to all. It can be really catastrophic when a true malignant disease sample which in this case in the minority data sample is misclassified. The class imbalance problem is generally incurred in the diagnosis fields such as medical diagnosis [108, 109], fraud detection [110, 111], intrusion detection [112, 113], bioinformatics [114], data gravitation [115] and finance risk management [116]

A path to counteract such situation is by preprocessing the datasets prior to feeding it as input to the classifier network. Most commonly used preprocessing methods for such kind of issue are over-sampling, under-sampling and ensemble learning. A wide range of survey of all the preprocessing methods have been done by many researchers [106, 117, 118]. The oversampling method is mostly concentrated upon in this research work. A widely used oversampling method which has been used in a lot of practical applications is the SMOTE method (Synthetic Minority Over-sampling Technique) introduced by Chawla et al [119]. A series of improvement to SMOTE has been carried out by many researchers from the time it is introduced [120–127]. Another approach to tackle imbalanced dataset is the multiple re-sampling method [128].

In most of the SMOTE related oversampling, the amount of oversampling done for each minority data sample is fixed. This means that the over sampling is done based on each minority data sample, i.e., if the amount of oversampling is 200% then each minority data sample generates two synthetic data. This approach at the end produces 200% of the whole minority

dataset. In this approach, the oversampling is done with respect to the whole minority dataset instead of each minority data sample. This means that for 200% oversampling of the minority dataset, the amount of generation of synthetic data sample for each minority data sample varies individually but in the end the total amount of oversampling increases up to 200% of the initial count of minority dataset. This approach in processed by assigning some weights to each of the minority data sample based on its Euclidean distance from rest of the minority data samples. The closer the minority data sample, i.e., the lesser the Euclidean distance, the more is the generation of synthetic data for that particular minority data sample. This modified method is named as Weighted-SMOTE as weights are assigned to each minority data sample for the generation of a particular number of synthetic data. The performance of Weighted-SMOTE method is calculated using some of the real world datasets those are widely used. These results are compared with the performance of SMOTE algorithm in order to make out the difference in performance between both of them. The performance is checked for a classifier network which is fed with the input dataset preprocessed using both SMOTE and Weighted-SMOTE individually. The performance of the network classifier is also calculated without any preprocessing of the dataset in order to check the credibility of both the oversampling algorithms. A k-fold cross validation approach is undertaken with a k value of ten and the final performance is averaged out of these ten folds.

This chapter deals with imbalanced datasets where the ratio of number of majority sample is way too much than the number of minority samples. This chapter does not deal with dimensionality reduction and the main focus is to have a biased learning by the classifier using a balanced dataset. Dimensionality reduction can be done once the dataset is balanced where the less variant attributes are discarded.

6.2 Synthetic minority over-sampling technique (SMOTE)

SMOTE algorithm is proposed to counteract the imbalanced dataset problem for classification [119]. It synthesizes new instances of the minority class by operating in the "feature space" rather than the "data space". This is an oversampling algorithm in which each minority data generates *N*% of synthetic data. The percentage increase in the minority data should be in such a way that it is comparable with the number of majority data. This increase in instances of the minority data expands the decision reasons for it in the classifiers.

In this algorithm, some of the parameters are initialized at the beginning. These parameters are the amount of oversampling (*N*%) that each of the *T* minority class samples need to undergo and *k* value of the *k* nearest neighbor of a particular minority class sample. Based on the amount of oversampling, the selection of one of the *k* nearest neighbor is chosen. For example, if the amount of oversampling is 300% and *k* value is 5, then randomly three out the five nearest neighbors are chosen for oversampling. After this initialization, a minority class sample is chosen and its *k* nearest neighbors out of the rest of the minority class samples is found. One among the *k* nearest neighbors of the minority class sample is randomly chosen. The first feature of this selected k nearest neighbor minority class sample is subtracted from the initially chosen minority class sample's first feature. This difference in value is multiplied with a gap value which is any random number between 0 and 1. This multiplied value is added to the initial minority class sample's first feature which is called as a synthetic data for it. This process is carried out for all the other features of a particular minority class sample which generates a row of synthetic sample for a particular minority class sample. In order to oversample it for *N*%, this process is carried out for a rounded value of (*N*/100) to its nearest integer. This generates *N*% of oversampling of a single minority class sample. This procedure is carried out for all the *T*

minority class samples which finally results in *N*% oversampling of all the minority class samples. In case the *N*% is less than 100%, then the *T* minority class samples are randomized so that only *N*% of *T* are considered for oversampling and *N* equals to 100.

6.3 Weighted Synthetic minority over-sampling technique (WSMOTE)

The weighted SMOTE method is an oversampling method which assigns weights that decides the number of new synthetic data which needs to be generated using SMOTE for an individual minority data. This is a slight modification to the original SMOTE where each of the minority data generating equal number of synthetic data. The weighted SMOTE method uses the Euclidean distance of each minority data sample with respect to all the other minority data samples in order to produce a weight matrix. This weight matrix along with the total percentage of synthetic data generation produces the SMOTE generation matrix which ultimately gives the number of synthetic data which needs to be generated for specific minority data sample as shown in Fig. 6.1.

$$[\boldsymbol{SMOTE\ Generation\ Matrix}]_{T\times 1} = \frac{\boldsymbol{N}\ \times\ \boldsymbol{T}}{\mathbf{100}}\ [\boldsymbol{Weight\ Matrix}]_{T\times 1} \qquad \textbf{(6.1)}$$

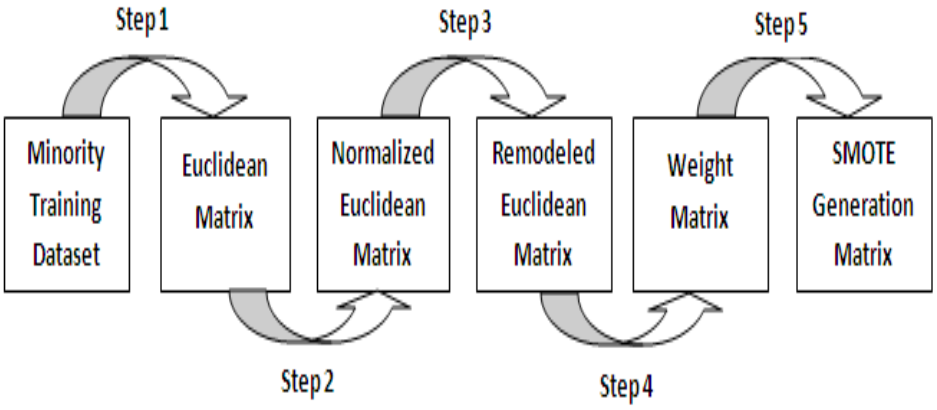


Figure 6.1: Block diagram of SMOTE Generation matrix

The steps involved in Weighted-SMOTE are as follows.

1. The minority training dataset is considered which contained *T* number of samples and each sample with *C* number of features. The Euclidean distance (ED) of each of the *T* minority data samples are calculated with respect to all the other minority data.

$$\mathbf{ED_i(m_i, m_j)} = \sqrt{\sum_{\mathbf{k=1}}^{\mathbf{C}} (\mathbf{m_{i,k}} - \mathbf{m_{j,k}})^2} \quad \textbf{(6.2)}$$

   Here, $i = [1, 2, \ldots, T]$ and $j = [1, 2, \ldots, T]$ and $j \neq i$. The ED for all the minority data are calculated and stored in a column matrix $ED = [ED_1, ED_2, \ldots, ED_T]^{'}$.

2. This ED matrix is then normalized using the maximum of the ED (ED_{max}) and the minimum of the ED (ED_{min}) and called as normalized ED matrix (NED). Normalization is done to map the numbers within a range 0 and 1.

$$\boldsymbol{NED_i} = \frac{\boldsymbol{ED_{i-}ED_{min}}}{\boldsymbol{ED_{max-}ED_{min}}} \quad \textbf{(6.3)}$$

3. The NED matrix is then modified to a remodeled normalized ED matrix (RNED). RNED matrix depicts that the lesser the ED of a minority data, the more share it gets to generate the synthetic data out of the total percentage of synthetic data (N%) that needs to be generated . RNED matrix is calculated by subtracting the normalized ED value of each minority data from the sum of all the normalized ED values.

$$[\boldsymbol{RNED}]_{\boldsymbol{T\times 1}} = \boldsymbol{sum(NED)} - [\boldsymbol{NED}]_{\boldsymbol{T\times 1}} \quad \textbf{(6.4)}$$

4. Finally, the weight matrix is calculated by finding each minority data share fraction with respect to the total sum of the shares in the RNED matrix.

$$[\boldsymbol{Weight\ Matrix}]_{T\times 1} = \frac{[\boldsymbol{RNED}]_{T\times 1}}{\boldsymbol{sum(RNED)}} \quad \textbf{(6.5)}$$

5. This weight matrix is used to find the SMOTE generation matrix.

This is explained with a simple example with T=5, N = 500% with the ED calculated as in Table 6.1

Table 15: Example using Weighted SMOTE

T	**1**	**2**	**3**	**4**	**5**
ED	2	1	5	3	4
NED	0.25	0	1	0.5	0.75
RNED	2.25	2.5	1.5	2	1.75
Weights	0.225	0.25	0.15	0.2	0.175
# SMOTE Generation	6	6	4	5	4

Table 6.1 shows that each minority data sample generates different number of synthetic minority data ranging from 4 to 6 instead of every minority data sample generating 5 synthetic data. This also shows that the smaller the ED, the larger share of synthetic data generation is assigned for that particular minority data sample.

6.4 Experimental Procedure

6.4.1 Experiment

To understand the performance of the proposed algorithm, some real world datasets shown in Table 6.2 are considered for analysis. All these datasets are converted to binary dataset with minority class samples and majority class samples. This approach can also be carried over for multi class classification problem considering multiple binary classes. These datasets have a

wide range of difference in the number of samples in the minority and majority classes. The ratio of minority to majority class samples is varied from 1:9 to 1:42.

6.4.1.1 Datasets

I. E coli dataset: This dataset contains the data regarding the protein localization sites for *Escherichia coli.* This dataset contains 8 attributes along with the class name. The dataset is segregated into a total of 8 classes with each class denoting one of the 8 protein localization sites. This chapter deals with binary imbalanced dataset problems, inner membrane cleavable signal sequence (imU) is considered as the minority class where as all the others are considered as majority class. Finally, the minority to majority class sample ratio is 35:301 which approximately is 1:9.

II. Abalone dataset: This dataset is used to find the age of the abalone based on 4177 samples and each sample consisting of seven features. The age of the abalone varies from 1 to 29. Hence, the total number of classes in this dataset is 29. For experimental purpose, class 9 is considered as the minority class with 42 samples and the class 18 is considered as the majority class with 689 samples. The ratio of minority to majority class sample in this dataset is 1:16.

III. Wine Quality dataset: This dataset classifies the white wine quality which ranges from 0 to 10 containing 4898 samples with each sample having 11 attributes. Class 8 is considered as minority class with 175 samples and the rest 4723 samples as majority class. The ratio of minority to majority class sample in this dataset is 1:27.

IV. Yeast dataset: This dataset classifies the localization position of protein in yeast with 1484 samples and each sample consisting of 8 attributes. Each sample can be classified to any one of the 10 localization site. ME2 (membrane protein, uncleaved signal) is

treated as minority class sample with 51 samples and the rest 1433 samples as majority class. The ratio of minority to majority class sample in this dataset is 1:28.

V. Mammography dataset: This dataset has 260 calcifications out of 11183 samples. These samples have 6 attributes in each sample. It is important that most of the 260 samples should be classified correctly. Hence, these 260 calcifications are considered as minority class samples where as the rest 10923 samples are considered as majority class samples. The ratio of minority to majority class sample in this dataset is 1:42.

Table 16: Dataset Distribution

Dataset	**Minority class**	**Majority class**	**#Minority class samples (T)**	**#Majority class samples (M)**	**Ratio (T/M)**
Ecoli	imU	Remainder	35	301	1:9
Abalone	9	18	42	689	1:16
Wine Quality	8	Remainder	175	4723	1:27
Yeast	ME2	Remainder	51	1433	1:28
Mammography	Calcifications	Non calcifications	260	10923	1:42

All these datasets are available in the University of California at Irvine (UCI) Repository[1]. The sample which has any missing attribute is deleted from the dataset.

6.4.1.2 Classifier Network

The classifier network used in this chapter is a multi layer perceptron (MLP) network which is trained by using the back propagation algorithm using Levenberg-Marquardt

[1] http://www.ics.uci.edu/~mlearn/

optimization method. The number of nodes in each hidden layer is equal to the number of attributes present in their respective datasets.

6.4.2 Procedure

Each dataset follows a specific procedure as shown in Fig. 6.2 before feeding it as a classifier input along with the machine learning algorithm. Each dataset is initially segregated into minority dataset and majority dataset based on the class label of each sample in the dataset. Afterwards, the minority dataset is divided into two sets, one with 10% and other with 90% of the minority dataset. In the same way, the majority dataset is also divided into two sets, one with 10% and the other with 90% of the majority dataset. This 10% of minority dataset and 10% of majority dataset are clubbed together to form the test dataset. The other 90% of minority dataset and 90% of majority dataset are clubbed to form the training dataset. Now this training dataset which contains both majority and minority dataset is fed as input to the classifier network. This classifier network is made to learn using a machine learning algorithm. After the classifier network is made to learn, the test dataset is used to check the performance of the trained classifier network. A *k*-fold cross validation with *k* value equal to 10 is applied and this procedure is made to run for 10 folds. The final performance of the classifier network is calculated as the average of the 10 folds. The recall and F-measure of the minority class samples are observed thoroughly. This is because these two measures explain clearly about the performance of the classifier network towards the minority class.

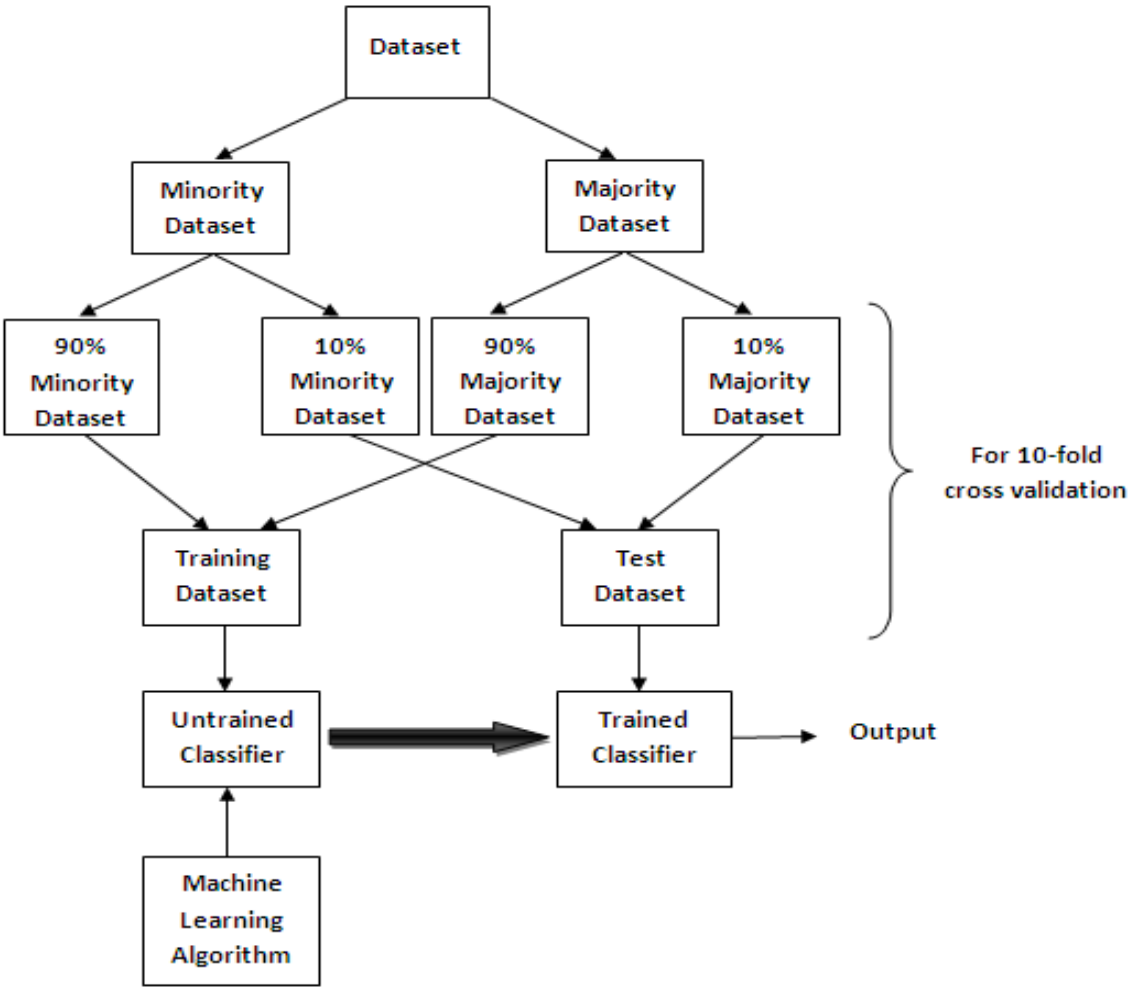


Figure 6.2: Preprocessing procedure of a dataset

6.5 Observation

The five datasets mentioned in Table 6.2 are used to analyze the performance of artificial neural network (ANN) multilayer perceptron in three different ways. In the first case, the performance of the classifier network is calculated when the dataset is directly fed as input to the ANN classifier without any preprocessing of data and it is denoted as ANN. In the second case, the performance of the classifier network is calculated when the dataset underwent SMOTE algorithm before being fed as input to the ANN classifier. This particular approach is denoted as SANN in this chapter. In the last case, the performance of the classifier network is calculated

when the dataset underwent Weighted SMOTE algorithm before being fed as input to the ANN classifier. This approach is denoted as WSANN in this chapter. The performance is also calculated for different values of the amount of oversampling (N) such as 100%, 200%, 300%, 400% and 500%.

For the Ecoli dataset, the WSANN gave better results compared to SANN and ANN for recall as well as F-measure in all the five values of N as shown in Figs. 6.3 and 6.4. The recall comparison depicted in Fig. 6.5 shows that WSANN gives better result for Abalone dataset than others in all the five N values. However, the F-measure for N equal to 200 for WSANN is marginally low compared to SANN as shown in Fig. 6.6. In case of Wine quality dataset, the recall and F-measure values for WSANN are nearly equal to or slightly better than that of SANN as shown in Figs. 6.7 and 6.8. The recall and F-measure values for the Yeast dataset using WSANN is comparably better than both SANN and ANN in all the five cases of oversampling values as shown in Figs. 6.9 and 6.10. In Fig. 6.11 which shows the recall values of the Mammography dataset, WSANN again showed better results compared to the other two whereas at 200% oversampling, SANN slightly underperformed compared to ANN. Similarly, for the same dataset, the F-measure of WSANN is always better than SANN and ANN whereas ANN slightly over performed compared to SANN at 200% and 300% oversampling as shown in Fig. 6.12.

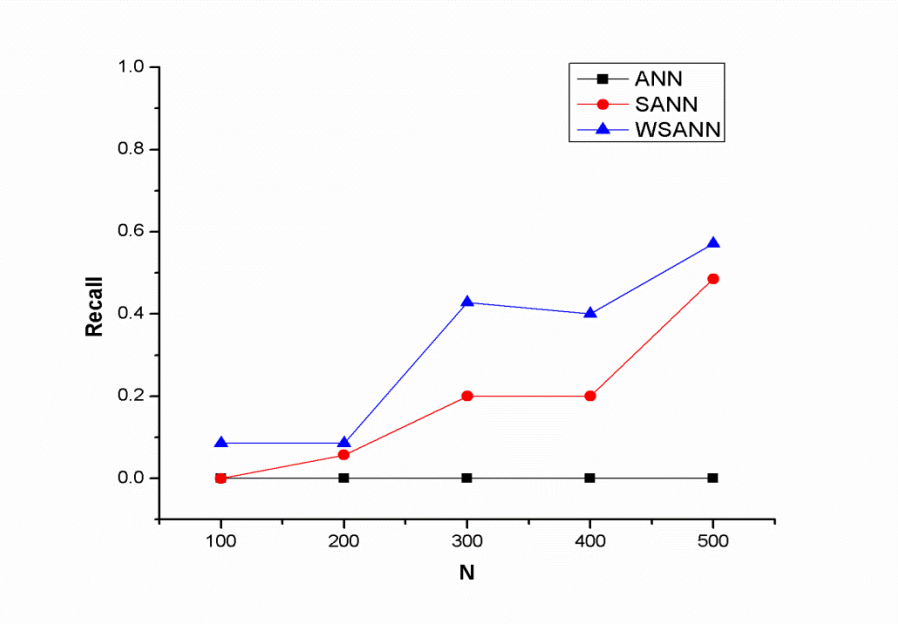


Figure 6.3: Comparison of Recall for Ecoli dataset

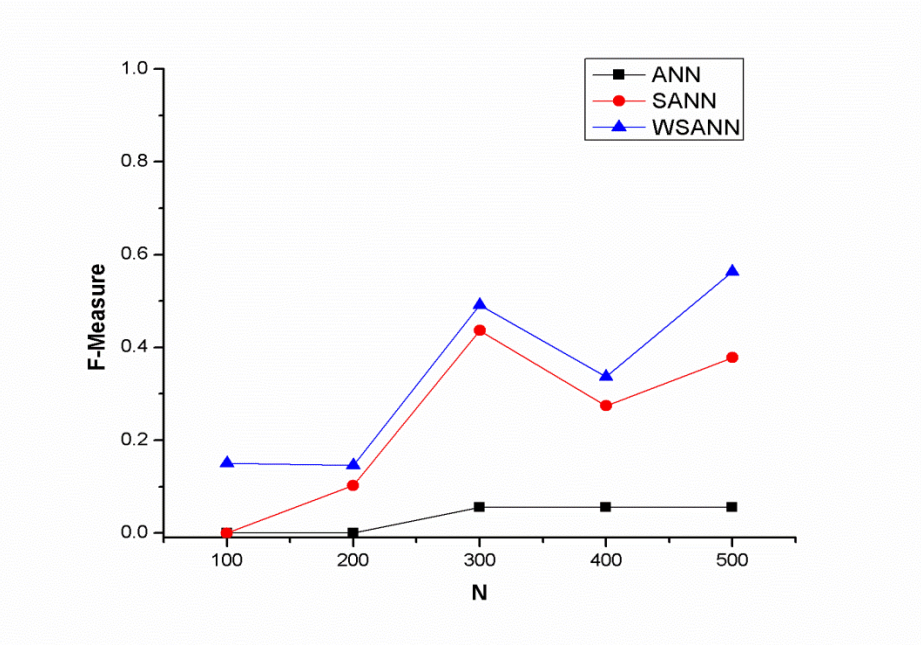


Figure 6.4: Comparison of F-Measure for Ecoli dataset

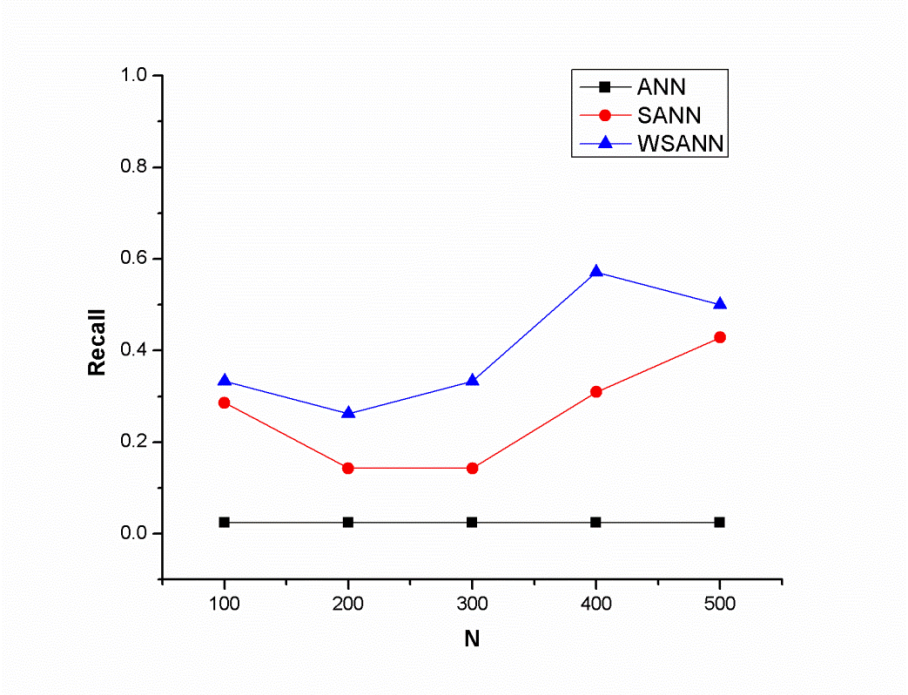


Figure 6.5: Comparison of Recall for Abalone dataset

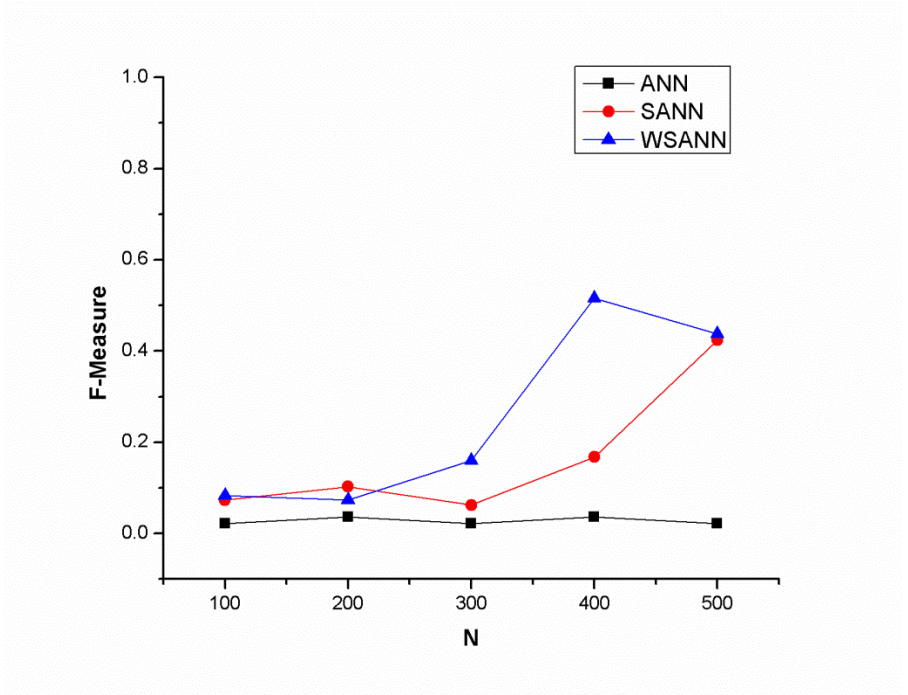


Figure 6.6: Comparison of F-Measure for Abalone dataset

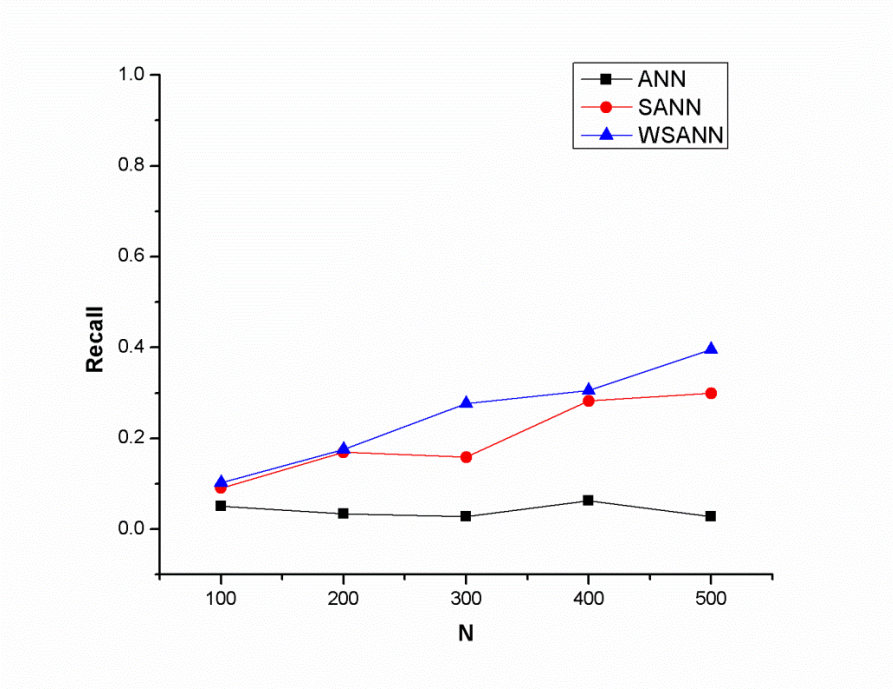


Figure 6.7: Comparison of Recall for Wine Quality dataset

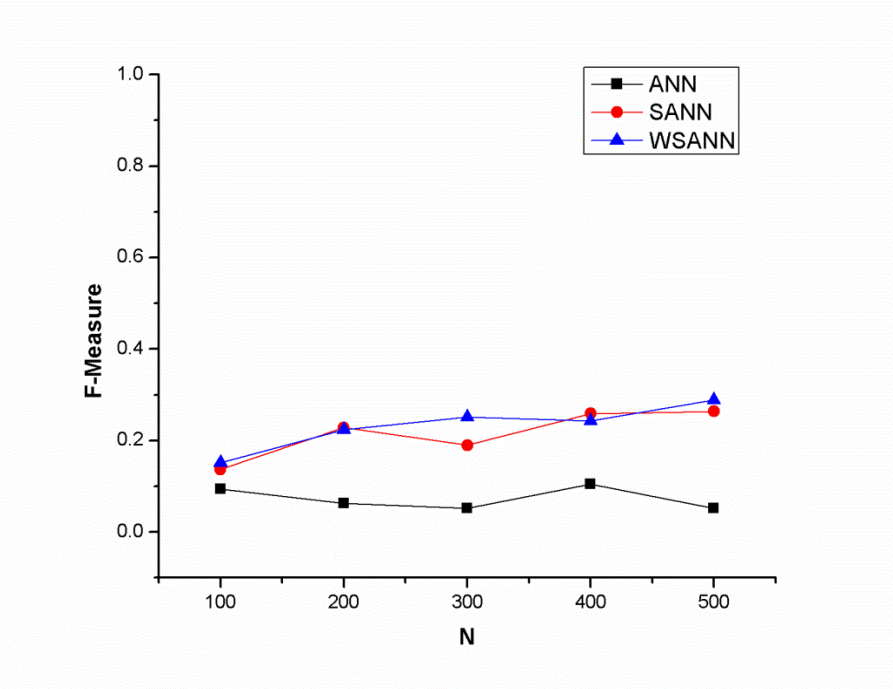


Figure 6.8: Comparison of F-Measure for Wine Quality dataset

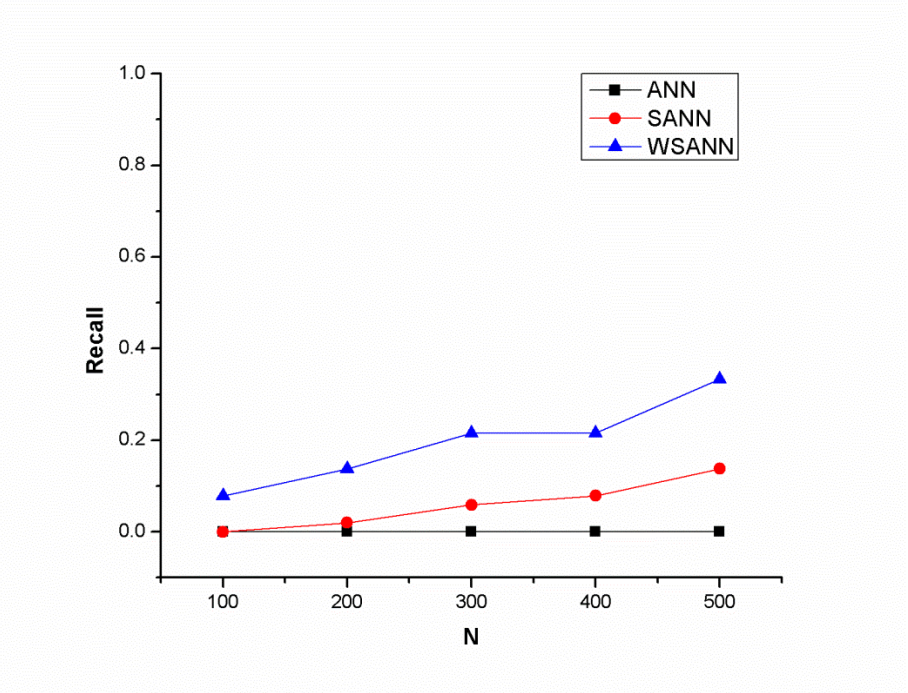


Figure 6.9: Comparison of Recall for Yeast dataset

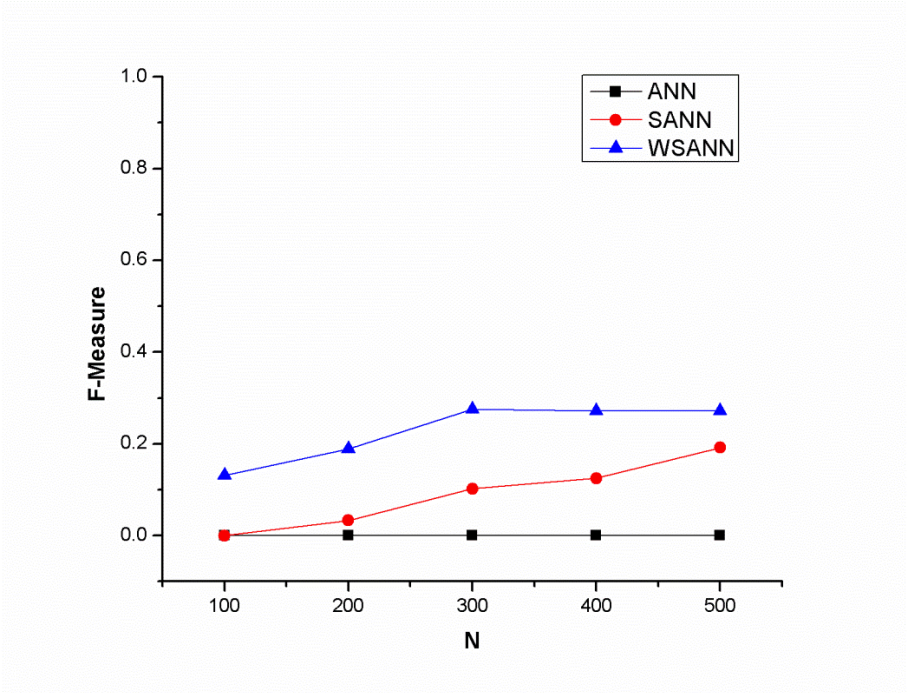


Figure 6.10: Comparison of F-Measure for Yeast dataset

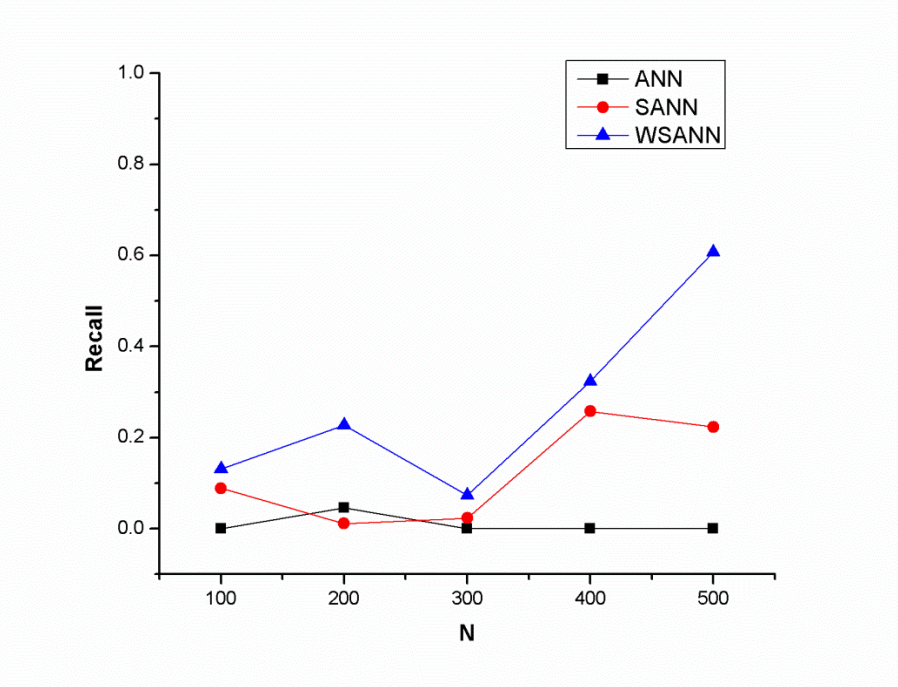


Figure 6.11: Comparison of Recall for Mammography dataset

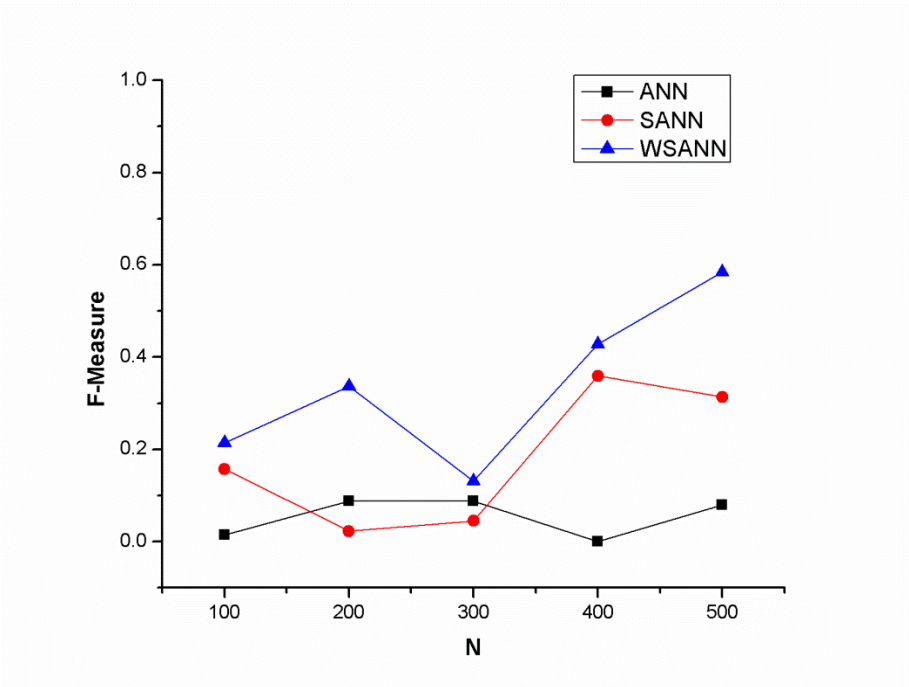


Figure 6.12: Comparison of F-Measure for Mammography dataset

6.6 Performance of WSMOTE and SMOTE for imbalanced dataset classification in a SFR

The application of both WSMOTE and SMOTE is verified using some of the malfunctions in a SFR. Two of these events that are subjected to imbalanced dataset are selected for testing purpose. The experiment is done in a 10-fold cross validation approach with the final performance metrics being the average of the 10 fold values. Here also, recall and f-measure are considered as the performance metrics and based on these two values comparison among all the three approaches is done. The three approaches are ANN classifier, ANN classifier with SMOTE and ANN classifier with WSMOTE.

The first event being the feed water speed reduction up to a certain percentage which occurs due to malfunctioning of the boiler feed pump located in the steam water system. This dataset contains data for the reduction in the feed water speed to 10%, 20%, 30% and so on till 90% where each of these percentage reductions is denoted as a class. Hence this dataset consisted of 9 classes and a total of 3451 samples. The features which are considered for each sample are the seven plant parameters considered in Section 4.4 of the present thesis. The reduction in the feed water speed to 90% is considered as the minority class containing 229 samples. Hence the ratio of minority samples to majority samples is nearly 1:14. Figures 6.13 and 5.14 show that at 200%, 400% and 500% oversampling of minority dataset, WSANN produces higher recall and f-measure respectively compared to SANN. In figure 6.13, SANN curve reaches a peak at 300, comes down at 400 and increases at 500. This is the behavior of ANN using SMOTE. This kind of fluctuation may occur in any kind of classifier which might have over fitted during the training phase. In the same figure, behavior of ANN using WSMOTE is in an increasing manner as the amount of oversampling increases which suggest the classifier

is not over fitted. Moreover, both SANN and WSANN produce better results in both the figures than ANN.

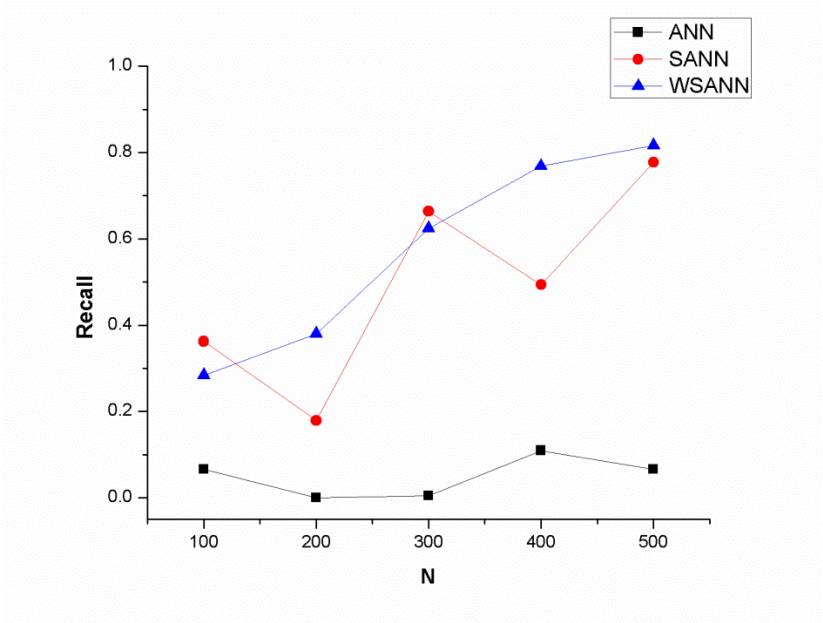


Figure 6.13: Comparison of Recall for feed water speed dataset

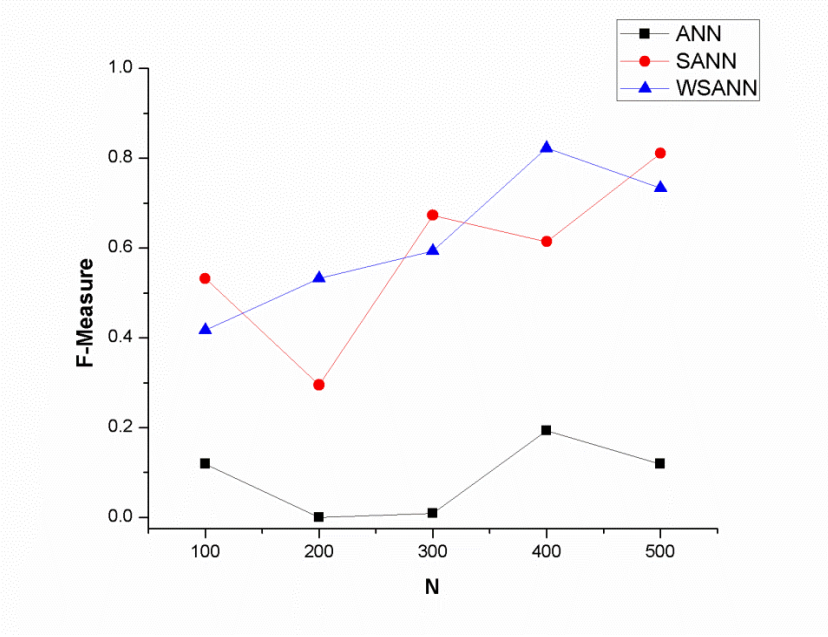


Figure 6.14: Comparison of F-Measure for feed water speed dataset

The second set of event from a SFR which has imbalanced dataset that is considered for validation of WSMOTE is the reduction in secondary sodium pump (SSP) speed due to its malfunctioning. The dataset considered for this set of event collected data for reduction in SSP speed by 10%, 20%, 30% and 40%. Here, reduction in SSP speed beyond 40% is not considered as the reactor goes to shutdown state. Hence, this dataset consisted of 4 classes and a total of 7785 samples. The seven plant parameters as mentioned in Section 4.4 are considered as features for each sample. The reduction in SSP speed by 40% is considered as the minority class with 708 samples. Hence, the ratio of minority samples to majority samples in this dataset is nearly 1:10. It is observed in Figs. 6.15 and 6.16 that both recall and f-measure are zero till the 400% oversampling of minority samples. Hence, the experiment is carried out till 800% oversampling for better analysis. It is quite evident from both the figures that both recall and f-measure are better in case of WSANN compared to SANN. Again, oversampling using SANN and WSANN produces better performance than ANN for classifying minority samples.

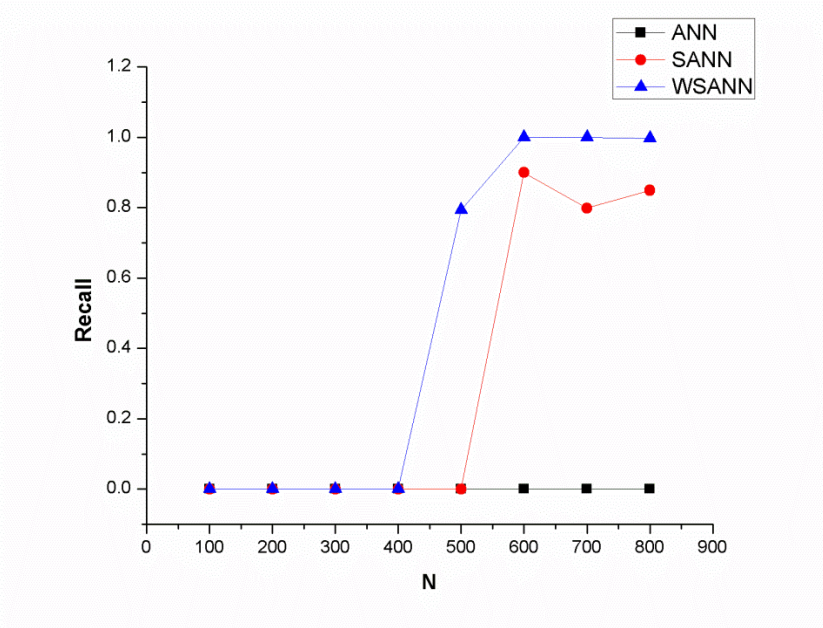


Figure 6.15: Comparison of Recall for SSP speed dataset

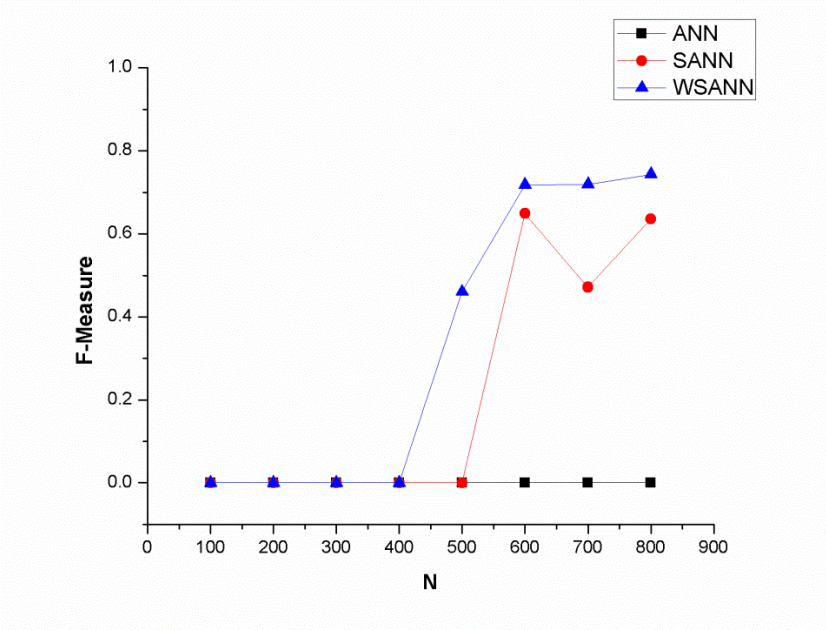


Figure 6.16: Comparison of F-Measure for SSP speed dataset

6.7 Summary

Imbalanced dataset problem and the classifier network learning from this imbalanced dataset are a real bottleneck for a machine learning researcher. One way of solving this issue is by oversampling the minority data samples. Out of the various oversampling methods, SMOTE has gained a lot of attention. SMOTE works on the principle of equal amount of oversampling (say 500%) of each minority sample which eventually ends up at the same amount of oversampling of the minority dataset on a whole (i.e. 500%). WSMOTE, a modification to SMOTE which believes in an approach where eventually in order to achieve a particular amount of oversampling of the minority dataset (say 500%), each minority sample need not be oversampled at the same rate as in SMOTE. The amount of oversampling for each minority sample is done based on the weights. A higher weight is given to samples which are nearby. This means that WSMOTE populates the area with more number of synthetic samples where the

number of minority samples is more. This reduces the chances of a misclassification of a test sample belonging to the minority class. In this chapter, it is observed that

- WSMOTE uses Euclidean distance to formulate a weighted matrix for oversampling of minority samples unequally.
- Recall and F-measure are the performance metrics which help in analyzing the actual performance of a classifier for an imbalanced dataset.
- WSMOTE along with ANN as classifier produces higher recall and f-measure for the considered datasets from UCI Repository datasets for most of the oversampling percentages compared to SMOTE with ANN classifier.
- WSMOTE along with ANN classifier also performs better in terms of recall and f-measure in identifying the considered malfunction in SFR compared to SMOTE with ANN classifier.

7

DEVELOPMENT OF GRAPHICAL USER INTERFACE FOR PLANT MONITORING

The present chapter explains the importance of GUI and its role in reducing the information overloading on the operator of NPP. The development of a GUI for online monitoring of plant status is also discussed. Some of the developed GUI having varied supervised classification algorithms along with multiple features for experimentation and analysis purpose have been also showcased.

7.1 Introduction

The main control room (MCR) of a NPP consists of a number of consoles, panels, displays, annunciation systems, hooters, etc which guide the operator to monitor as well as control the plant. A filtered number of critical and important signals only are monitored at the MCR. During off-normal condition of the plant, the information subjected to the operator should be productive and reduced in number that avoids information overloading. Information overload occurs when the amount of input to a system exceeds its processing capacity [129]. Information overload increases the time required to make a decision and increases the confusion regarding the decision [130]. Consequently, when information overload occurs, it is likely that a reduction in decision quality will occur [131]. Hence, reduced information overloading on the operator during such imbalanced condition aids the operator in improving the decision making ability by providing quick and unambiguous decisions.

One of the ways to provide reduced information overloading on the operators is by producing well-crafted and user-friendly graphical user interface (GUI) which displays crisp and obligatory information during off-normal conditions. In a safety critical system like NPP, the requirement for faster prediction systems has sparked the usage of intelligent approaches. The coagulation of various artificial intelligent algorithms and GUI provides and intelligent way of interpreting and displaying data. These GUI produces a very easy interactive interface for experimenting on various supervised classification algorithms and finding the performance of the classifier. The inclusion of the various dataset transformation techniques explained in the previous chapters have also been considered in the GUI.

7.2 Salient Features of GUI

As the GUI concentrates on reduction of information overloading on the operator, there are some salient features that has been taken into consideration in developing a GUI. These salient features are as follows.

I. The interface must be simple, user friendly and understandable.

II. Each of the stages in the interface must be systematic and any unnecessary ambiguous display must be avoided.

III. Graphs, tables and charts need to be used where ever necessary in order to display results more clearly.

IV. Prominent font size including proper color combination should be used in the interface for denoting graphs, words, etc.

V. The interface must take care of all the possible error handling scenarios avoiding sudden collapse or seize of the interface.

VI. Continuous operation of the code for different datasets must be supportable.

7.3 GUI for fuzzy rule based classification system

The interface developed to experiment on the developed FRBCS mentioned in Chapter 2 is shown in Figs. 7.1 and 7.2 named as GUI-FRBCS classification tool. Both the figures depict a simple GUI which browses for a dataset stored in a database and gives defuzzified output of each sample. This defuzzified output is calculated based on the developed FRBCS for transient identification. The defuzzified output with 0 depicts that the input test sample represents the plant running in full power as shown in Fig. 7.1. The status of the plant is displayed in the textbox present on the top of the graph. Any value other than 0 represents that the respective sample represents a transient data. It is also evident from these figures that the color of the plot also has importance. If the plot is green in color it represents the plant in safe condition that is full power. If the plot is in red color it represents that plant is in an unsafe condition because of the occurrence of a transient. The mapping of the defuzzified output value with the transient is given in Table 7.1. Figure 7.2 shows that the defuzzified value has been stabilized at 1 and 4. It means the data that is fed belongs to one condensate extraction pump trip with stand by not taking over transient and both boiler extraction pump trip with stand by not taking over transient for the defuzzified values of 1 and 4 respectively. This interface helps in online monitoring of the plant for the considered five classes.

Table 17: Mapping of defuzzified value with transients

Defuzzified output	Class or Plant condition
0	Full Power
1	One condensate extraction pump trip with stand by not taking over
2	Both condensate extraction pump trip with stand by not taking over
3	One boiler extraction pump trip with stand by not taking over
4	Both boiler extraction pump trip with stand by not taking over

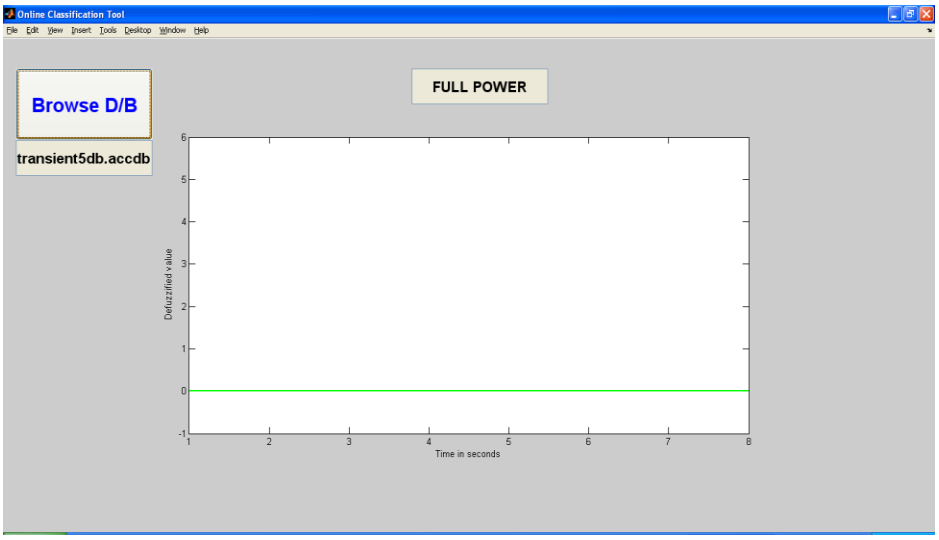


Figure 7.1: GUI-FRBCS classification tool for full power state

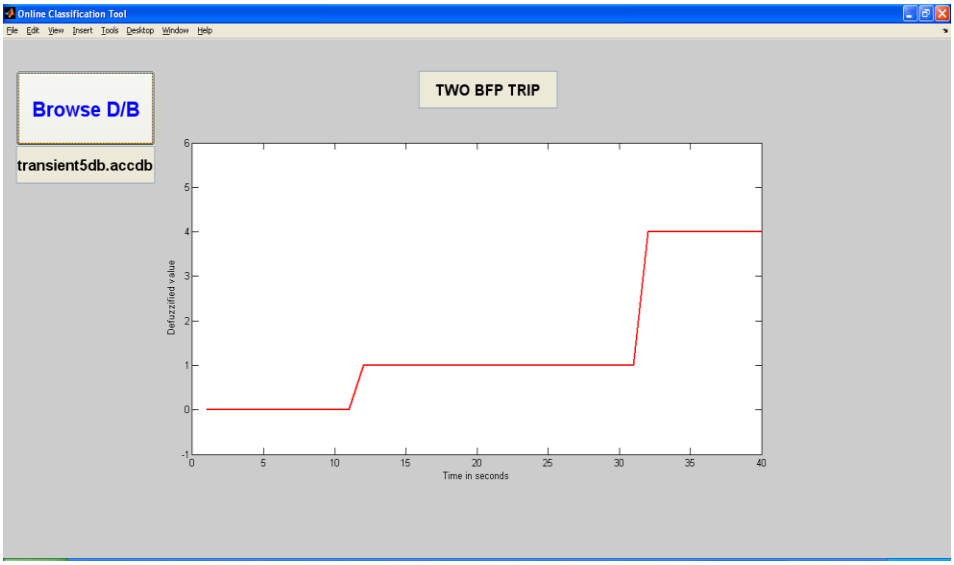


Figure 7.2: GUI-FRBCS classification tool for transients

7.4 GUI for machine learning based classification

Figure 7.3 shows the GUI which uses some of the very well known machine learning algorithms for classification problem. This GUI has an option to choose PCA for dimensionality reduction, if necessary, and analyze the performance of the considered classifier. Hence, this GUI is referred to as GUI-ML-PCA classification tool. The GUI basically is segregated into four segments as follows.

I. Machine learning (ML) algorithms segment

II. Training segment

III. Testing segment

IV. Results segment

The first segment of GUI-ML-PCA is dedicated to selection of the machine learning algorithm. The algorithms which have been stated for selection are k-nearest neighbor, support vector machine and artificial neural network as the classifier. The selection of the ANN algorithm is done based on any of the different backpropagation algorithms. These algorithms along with the expansion of the abbreviations are mentioned in Table 7.2. Any one of these algorithms can be chosen at a time. Only one hidden layered neural network is considered in this GUI. The number of nodes in this hidden layer can be fed by the user in the editable text box present in the bottom right in this segment. Once this stage is selected, the user can proceed to the second stage.

The second stage is the training of the chosen classifier in GUI-ML-PCA. This segment has a provision to browse for the training dataset which would be fed to the chosen classifier. The inclusion of the dimensionality reduction using PCA has made this GUI more fascinating. This is an optional choice where implementing or not implementing this approach is dependent on the user by selecting the appropriate radio button in this segment. The selection of the number of principal components on approving to implement PCA in the training phase is mandatory. This gives an option to the user to analyze for various factor numbers in order to test the performance of the PCA based classifier for each factor number. There are a few buttons in this segment which implement the intended action on clicking it. This stage trains the chosen classifier on the selected training dataset with or without the implementation of PCA.

The third stage of GUI-ML-PCA is dedicated to testing of the trained classifier. It is done by browsing through a test dataset. PCA is taken care of using the inputs from the previous stage. Hence, there is no need editable text box present in this segment for this purpose. The overall prediction accuracy of the classifier is displayed in a text box in this segment once the "TEST" button is pressed.

The final segment of GUI-ML-PCA shows a detailed result on the performed experiment. Among all the buttons in this segment, the test accuracy, confusion matrix and receiver operating characteristics (ROC) primarily give an insight on the actual performance of the selected classifier. The table segment in this segment also shows the data in the training stage and the test stage.

Table 18: Full form of the Backpropagation algorithms

Abbreviation	**Back propagation algorithm**
BR	Bayesian Regularization
RBP	Resilient Backpropagation
GDM	Gradient Descent with momentum backpropagation
GDA	Gradient Descent with adaptive learning rate backpropagation
GDMA	Gradient Descent with momentum and adaptive learning rate backpropagation
SCG	Scaled conjugate gradient back propagation
CGB	Conjugate gradient backpropagation with Powell-Beale restarts
CGF	Conjugate gradient backpropagation with Fletcher-Reeves updates
CFP	Conjugate gradient backpropagation with Polak-Ribiére updates
LM	Levenberg-Marquardt backpropagation
QN	BFGS quasi-Newton backpropagation
OSS	One-step secant backpropagation

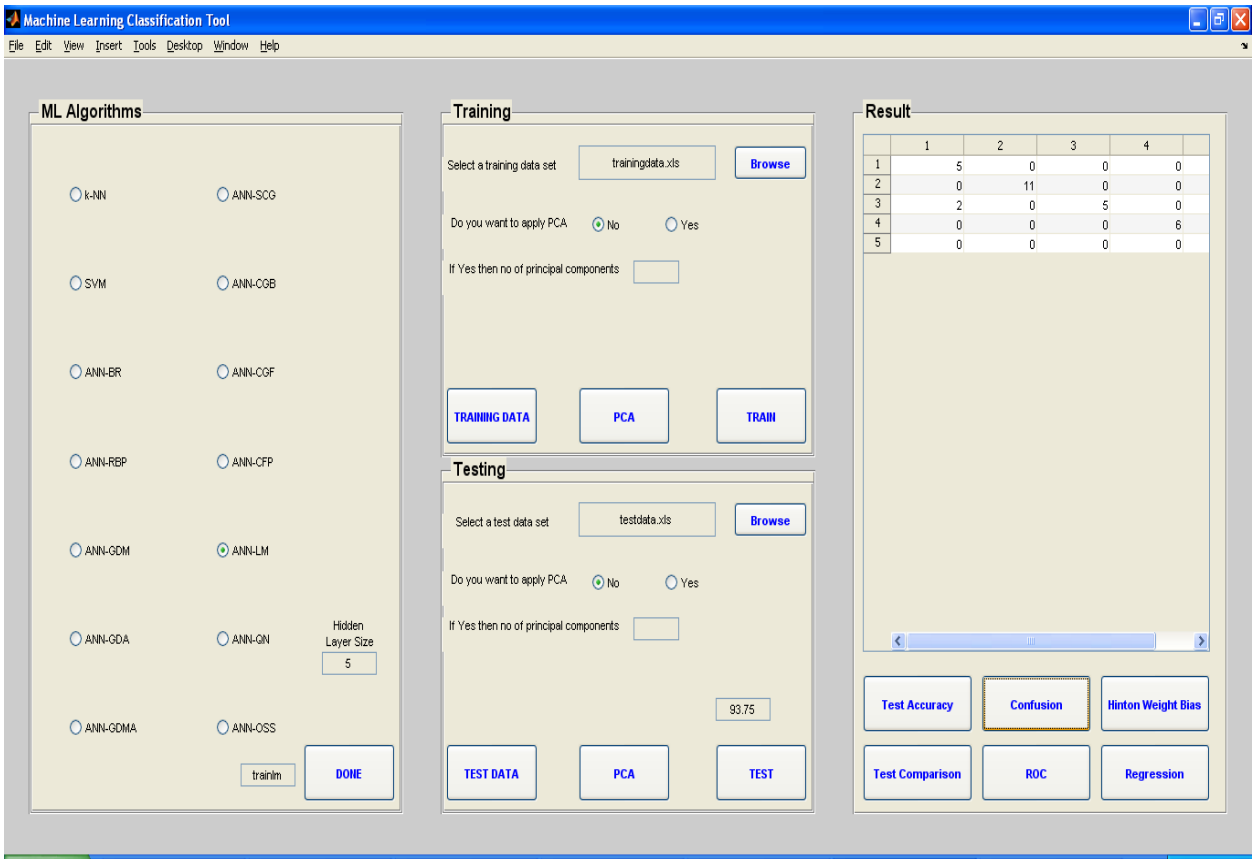


Figure 7.3: GUI-MLC-PCA classification tool

7.5 Modified GUI for machine learning based classification

Figures 7.4 and 7.5 represent the modified version of GUI-ML-PCA which includes training dataset reduction approach. Hence it is called as GUI-ML-TDR-PCA classification tool. The addition of TDR to this interface includes the selection of $ED_{cut\text{-}off}$ using any one of the two methods as delineated in Chapter 3. In order to give more clarity on the confusion matrix, the metrics deduced from it has been displayed in two ways. The performance metrics explained in Chapter 1 such as sensitivity, fallout, miss rate, specificity and accuracy have been used and their numeric values are displayed as shown in Fig. 7.4 and named as GUI-ML-TDR-PCA-1 classification tool. This has been shown in both the training and the test segment. Figure 7.5 shows the confusion matrix instead of respective performance metrics and is named as GUI-ML-

TDR-PCA-2 classification tool. Another improvement of this classification tool over the previous one is the inclusion of ANFIS classifier in the algorithm list.

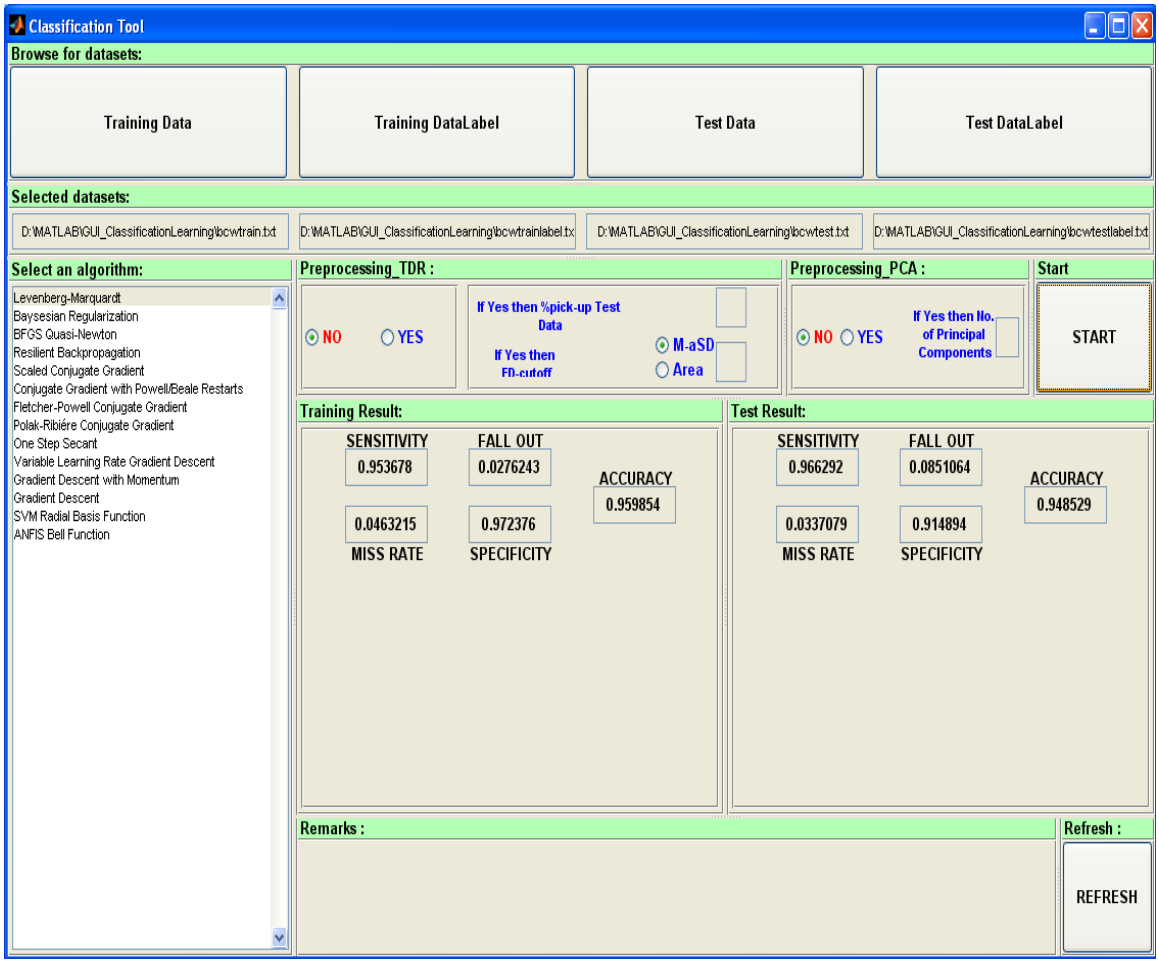


Figure 7.4: GUI-ML-TDR-PCA-1 classification tool

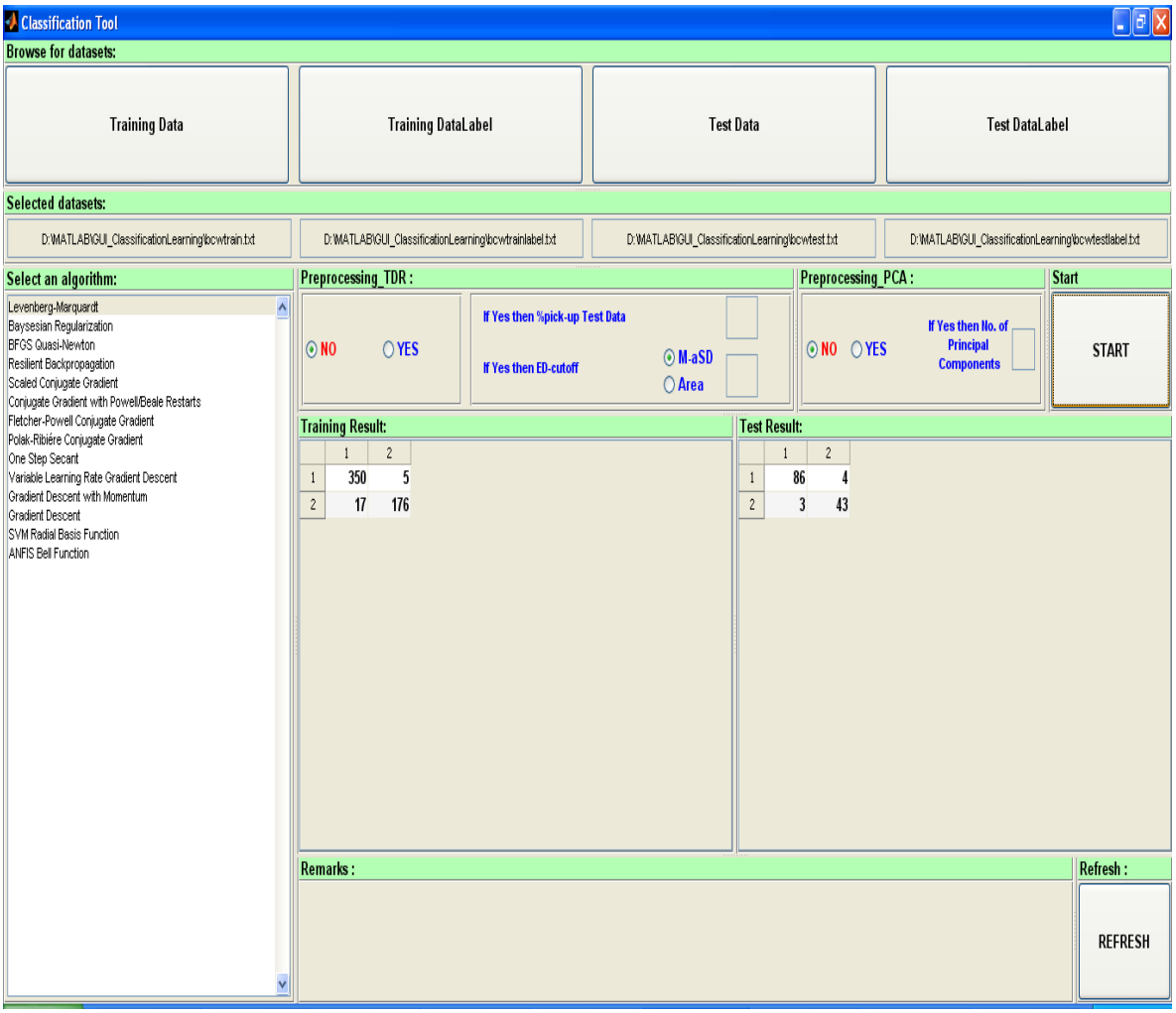


Figure 7.5: GUI-ML-TDR-PCA-2 classification tool

7.6 Summary

A user-friendly GUI leads to reduction in information overloading on anyone who is using it. The necessity of displaying only the important and relevant information reduces the time of reaction which is due to the avoidance of confusion. This eventually leads to better decision making ability. The usage of such simple yet informative GUI can also be used for online monitoring of parameters for knowing about the plant status. Experimentation and analysis of various algorithms in one screen becomes easier. From the developed GUI, it is observed that

- The comparative study on these algorithms based on the performance of the classifier can be done instantly.
- GUI-FRBCS classification tool can be used for online monitoring of plant status for the considered events.
- GUI-ML-PCA classification tool can be used for trying out various modifications to the number of nodes in the hidden layer and algorithm selected for a particular classifier.
- It also includes a feasibility check on various factor numbers for PCA for a dataset for analysis purpose.
- GUI-ML-TDR-PCA classification tool can be used for both TDR and PCA approaches for dataset transformation for a considered dataset using a selected supervised classification algorithm.

8

SUMMARY AND SCOPE FOR FUTURE WORK OF THE THESIS

The present chapter summarizes the different conclusions from the research work carried out in the field of supervised classification algorithms based on dataset transformation in monitoring nuclear power plant events. It also provides the scope for future work in this genre for monitoring nuclear power plant events and also in general.

8.1 Summary of the thesis

The thesis focuses on online monitoring of events and development of dataset transformation methods for various supervised classification algorithms. The main objective of any classifier is to achieve the highest possible prediction accuracy. Using dataset transformation, it has been observed that in addition to increased prediction accuracy, reduction in computational cost can also be achieved. The reduction in computational cost may be by reducing training time or training samples or number of features. The motivation behind this research work has been due to the huge dataset and the amount of time each classifier takes to get trained. Moreover, there is no guarantee that the final accuracy of an algorithm is better than the previous algorithm. Hence, it is better that a reduced computational cost helps in trying a variety of modifications.

Online monitoring of occurrence of transients in NPP is developed using FRBCS. It is observed that a properly prepared rule base and membership functions using expert's knowledge

for a FRBCS are feasible for transient identification in SFR. It is noted that for NPP which has a cycle time of 200 milliseconds, a waiting time or response time of a few seconds can give conclusive evidence on the occurrence of a particular transient. Moreover, it is informed earlier that the considered events are not fast occurring events and take some seconds to minutes in order to trip the reactor. It is also observed that for the considered transients in the steam water system of a SFR, deaerator level and change in deaerator level as the input features produced significantly lower response time for four out of five transients compared to other considered plant parameters. It is found that increase in the number of input features does not necessarily increase the performance of the classifier. Hence, judiciously implementing feature selection for a fuzzy system can lead to a more accurate system whose interpretability is not penalized much.

A novel approach for reduction in training samples termed as training dataset reduction approach is examined. This approach is implemented to reduce the computational cost of training phase of a classifier without compromising on the classification accuracy. This reduction is done by creating a hypothetical boundary based on a cut-off Euclidean distance. The cut-off Euclidean distance is calculated using either Mean-αStandard deviation method or area selection method. For the considered real world datasets, it is found that for a 5% pick-up dataset, Mean-αStandard deviation based TDR approach provides acceptable results for $\alpha=1$ (for considerable amount of data samples) and $\alpha=2$ (for high amount of data samples) and for area selection based TDR approach gives satisfactory results for 60%. For datasets with higher number of classes, the pick-up dataset percentage has to be increased till at least one sample from each class gets selected in the pick-up dataset. The inference from the real world datasets is tested on some of the transients in a SFR. It is found that Mean-αStandard deviation based TDR approach gives better and acceptable results than area selection based TDR approach with a positive difference

in the classification accuracy along with a maximum of about 70% reduction in training samples. TDR approach does not produce improved result for the dataset consisting of less training samples. In such cases, classifier without the implementation of TDR approach proves to be a better option for achieving higher classification accuracy. These results have sparked the importance and feasibility of TDR approach in NPP and also in other domains.

Another approach for reducing the computational cost of the training phase of a classifier is by the implementation of dimensionality reduction. In this approach, the most important features are kept intact dumping the others. Principal component analysis is an algorithm which can be used for dimensionality reduction. It is found that on usage of PCA beyond a certain factor number produces better performance than the non usage of PCA. The selection of this cut-off factor number is found using scree-plot approach which arranges the factor number based on the descending order of their eigen values of the covariance matrix. Hence, the factor number with a steep drop or an eigen value of nearly zero is discarded. The performance measure for this approach is the area under the receiver operating characteristics curve (AUC). This approach is used for classifying some of the malfunction events in a SFR using an ANFIS classifier. It is concluded that beyond a factor number 4 for a seven featured dataset of SFR malfunction event, PCA along with MANFIS produces higher AUC than MANFIS without the usage of PCA. A comparison on the performance of a multiple-ANFIS and single ANFIS is also carried out. From this comparison it is inferred that PCA based MANFIS produces better AUC than PCA based single ANFIS. Hence, dividing an event to sub events like label-1event or primary event and label-2 event or ancillary event and then classifying it using MANFIS classifiers gives better performance measures compared to a single ANFIS classifier.

A very commonly used algorithm to overcome the problem of imbalanced dataset is the oversampling method. SMOTE is one of the very popular oversampling algorithms. In this algorithm, the oversampling of the minority data samples is done in equal numbers. A modification to this approach is the WSMOTE which uses weighted approach to oversample the minority samples. In this algorithm, the oversampling of the minority data samples need not be equal. A comparison on the performance of a classifier using SMOTE and WSMOTE reveals that for the considered real world datasets, WSMOTE produced higher recall and f-measure. Recall and f-measure are the two metrics which give a clarity regarding the actual performance of a classifier for an imbalanced dataset. The comparison is done at different percentage of oversampling. This approach is tested for classifying some of the malfunction events in a SFR. Again, it is observed that WSMOTE produced better recall and f-measure for most of the oversampling percentages compared to SMOTE. It is quite obvious that WSMOTE and SMOTE based classifier performed better in all cases than using a classifier without the implementation of any of these oversampling techniques.

The usage of a user-friendly GUI for monitoring plant status without information overloading on the operator increases the decision making ability. Also various implementation using a variety of algorithms on a single interface helps in proper experimentation and analysis. GUI-FRBCS classification tool can be used for online monitoring of plant status for the considered events. Both GUI-ML-PCA classification tool and GUI-ML-TDR-PCA classification tool can be used to experiment on various modifications and implementation of datasets in various supervised classification algorithms. This reduces the time in writing code for each and every algorithm separately.

8.2 Scope for future work of the thesis

The work presented in the thesis emphasizes on the dataset transformation and significantly improves the classification algorithm of various supervised classification algorithms. Practically, the thesis proposed new and efficient algorithms for classification algorithm for normal datasets and imbalanced datasets. Based on the successful completion of various algorithms, the scope of future work is outlined as follows.

- In case of classification using FRBCS, the need to integrate other transient and malfunction to a single model is a challenging job. Yet, this would certainly bring everything into a single model.
- Again, in FRBCS, the feature selection is the key to its classification accuracy. Hence, a proper analysis on various other plant parameters for classifying all the other transients can produce the apt features for a set of transients.
- In case of TDR approach, a more generalized way of choosing the pick-up dataset and cut-off boundary can make the approach more usable in wide spread domain.
- PCA based ANFIS classifier is used for seven featured malfunction dataset in SFR. Hence, the feasibility and performance quality can be checked for a higher dimension dataset in order to prove its credibility in a more convincing manner.
- WSMOTE and SMOTE based classifier are compared for various oversampling percentages. Hence, there is not logic behind choosing a particular oversampling percentage which should definitely give better accuracy. There is a need to study on this aspect of selecting the oversampling percentage.
- For all these approaches, the testing has been done on few of the classification problems in restricted portions of a SFR. This study can be extended on to various other events in

other portions of a SFR in order to analyze the feasibility of such approaches in various distributed genres of a SFR.

REFERENCES

1. Deolalikar R (2008) Safety in nuclear power plants in India. Indian J Occup Environ Med 12:122–127.

2. Ram KS, Iyer K (1987) Safety of nuclear power plants. Sadhana 11:263–272.

3. Chetal SC, Balasubramaniyan V, Chellapandi P, et al (2006) The design of the Prototype Fast Breeder Reactor. Nucl Eng Des 236:852–860.

4. Jayanthi T, Seetha H, Narayanan KRS, et al (2011) Simulation and Integrated Testing of Process Models of PFBR Operator Training Simulator. Energy Procedia 7:653–659.

5. Jasmine N, Nawlakha R, Seetha H, et al (2011) Simulation of control logics for plant transition state for PFBR Operator Training Simulator. In: 2011 Int. Conf. Recent Trends Inf. Technol. ICRTIT. pp 380–384

6. Carlos Canedo Medeiros JA, Schirru R (2008) Identification of nuclear power plant transients using the Particle Swarm Optimization algorithm. Ann Nucl Energy 35:576–582.

7. Moshkbar-Bakhshayesh K, Ghofrani MB (2013) Transient identification in nuclear power plants: A review. Prog Nucl Energy 67:23–32.

8. Bloch I (2015) Fuzzy sets for image processing and understanding. Fuzzy Sets Syst 281:280–291.

9. Chanak P, Banerjee I (2016) Fuzzy rule-based faulty node classification and management scheme for large scale wireless sensor networks. Expert Syst Appl 45:307–321.

10. Karakış R, Güler İ, Çapraz İ, Bilir E (2015) A novel fuzzy logic-based image steganography method to ensure medical data security. Comput Biol Med 67:172–183.

11. Ferreira VH, Zanghi R, Fortes MZ, et al (2016) A survey on intelligent system application to fault diagnosis in electric power system transmission lines. Electr Power Syst Res 136:135–153.

12. Wei W, Mendel JM (1999) A fuzzy logic method for modulation classification in nonideal environments. IEEE Trans Fuzzy Syst 7:333–344.

13. Sanz JA, Fernández A, Bustince H, Herrera F (2010) Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. Inf Sci 180:3674–3685.

14. Jahromi MZ, Taheri M (2008) A proposed method for learning rule weights in fuzzy rule-based classification systems. Fuzzy Sets Syst 159:449–459.

15. Fernández A, Calderón M, Barrenechea E, et al (2010) Solving multi-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations. Fuzzy Sets Syst 161:3064–3080.

16. Fazzolari M, Giglio B, Alcalá R, et al (2013) A study on the application of instance selection techniques in genetic fuzzy rule-based classification systems: Accuracy-complexity trade-off. Knowl-Based Syst 54:32–41.

17. Ishibuchi H, Nakashima T (2001) Effect of rule weights in fuzzy rule-based classification systems. IEEE Trans Fuzzy Syst 9:506–515.

18. Sboev A, Litvinova T, Gudovskikh D, et al (2016) Machine Learning Models of Text Categorization by Author Gender Using Topic-independent Features. Procedia Comput Sci 101:135–142.

19. de Bruijne M (2016) Machine learning approaches in medical image analysis: From detection to diagnosis. Med Image Anal 33:94–97.

20. Lim S, Tucker CS, Kumara S An unsupervised machine learning model for discovering latent infectious diseases using social media data. J Biomed Inform.

21. Yang J, Honavar V (1998) Feature Subset Selection Using a Genetic Algorithm. In: Liu H, Motoda H (eds) Feature Extr. Constr. Sel. Springer US, pp 117–136

22. Veldhuizen DAV, Lamont GB (2000) Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. Evol Comput 8:125–147.

23. Ishibuchi H, Murata T, Türkşen IB (1997) Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. Fuzzy Sets Syst 89:135–150.

24. Karaboga D, Ozturk C (2009) Neural Networks Training by Artificial Bee Colony Algorithm on Pattern Classification. Neural Netw World 19:279–292.

25. Holden N, Freitas AA (2008) A Hybrid PSO/ACO Algorithm for Discovering Classification Rules in Data Mining. J Artif Evol App 2008:2:1–2:11.

26. Ma J, Jiang J (2011) Applications of fault detection and diagnosis methods in nuclear power plants: A review. Prog Nucl Energy 53:255–266.

27. Uhrig RE, Tsoukalas LH (1999) Soft computing technologies in nuclear engineering applications. Prog Nucl Energy 34:13–75.

28. Gottlieb C, Arzhanov V, Gudowski W, Garis N (2006) Feasibility Study on Transient Identification in Nuclear Power Plants Using Support Vector Machines. Nucl Technol 155:67–77.

29. Rocco S. CM, Zio E (2007) A support vector machine integrated system for the classification of operation anomalies in nuclear components and systems. Reliab Eng Syst Saf 92:593–600.

30. RaniPatra S, Jehadeesan R, Seetha H, et al (2012) Event Identification in Prototype Fast Breeder Reactor Subsystem using Artificial Neural Network. Int J Appl Inf Syst 4:1–9.

31. Santosh TV, Vinod G, Saraf RK, et al (2007) Application of artificial neural networks to nuclear power plant transient diagnosis. Reliab Eng Syst Saf 92:1468–1472.

32. Roverso D (2007) Intelligent normalisation for transient classification. Int J Nucl Knowl Manag 2:239–248.

33. Cheon SW, Chang SH (1993) Application of Neural Networks to a Connectionist Expert System for Transient Identification in Nuclear Power Plants. Nucl Technol 102:177–191.

34. Santosh TV, Srivastava A, Sanyasi Rao VVS, et al (2009) Diagnostic system for identification of accident scenarios in nuclear power plants using artificial neural networks. Reliab Eng Syst Saf 94:759–762.

35. Bartal Y, Lin J, Uhrig RE (1994) Nuclear power plants transient diagnostics using LVQ or some networks don't know that they don't know. In: 1994 IEEE Int. Conf. Neural Netw. 1994 IEEE World Congr. Comput. Intell. pp 3744–3749 vol.6

36. Marseguerra M, Zio E, Oldrini A, Brega E (2003) Fuzzy identification of transients in nuclear power plants. Nucl Eng Des 225:285–294.

37. Zio E, Baraldi P (2005) Identification of nuclear transients via optimized fuzzy clustering. Ann Nucl Energy 32:1068–1080.

38. Zio E, Baraldi P (2005) Evolutionary fuzzy clustering for the Classification of transients in nuclear components. Prog Nucl Energy 46:282–296.

39. Roverso D (2002) Plant diagnostics by transient classification: The ALADDIN approach. Int J Intell Syst 17:767–790.

40. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353.

41. Wang L-X, Mendel JM (1991) Generating fuzzy rules by learning from examples. In: Proc. 1991 IEEE Int. Symp. Intell. Control 1991. pp 263–268

42. Mendel JM, Mouzouris GC (1997) Designing fuzzy logic systems. IEEE Trans Circuits Syst II Analog Digit Signal Process 44:885–895.

43. Cordón O, del Jesus MJ, Herrera F (1999) A proposal on reasoning methods in fuzzy rule-based classification systems. Int J Approx Reason 20:21–45.

44. Riid A, Rüstern E (2014) Adaptability, interpretability and rule weights in fuzzy rule-based systems. Inf Sci 257:301–312.

45. Poonam, Tripathi SP, Shukla PK Uncertainty Handling using Fuzzy Logic in Rule Based Systems. Int J Adv Sci Technol 45:31–46.

46. Hamam A, Georganas ND (2008) A comparison of Mamdani and Sugeno fuzzy inference systems for evaluating the quality of experience of Hapto-Audio-Visual applications. In: IEEE Int. Workshop Haptic Audio Vis. Environ. Games 2008 HAVE 2008. pp 87–92

47. Kaur A, Kaur A (2012) Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System. Int J Soft Comput Eng 2:323–325.

48. Ghahramani Z (2004) Unsupervised Learning. In: Bousquet O, Luxburg U von, Rätsch G (eds) Adv. Lect. Mach. Learn. Springer Berlin Heidelberg, pp 72–112

49. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement Learning: A Survey. J Artif Int Res 4:237–285.

50. Yang L (2006) Distance Metric Learning: A Comprehensive Survey.

51. Weinberger KQ, Saul LK (2009) Distance Metric Learning for Large Margin Nearest Neighbor Classification. J Mach Learn Res 10:207–244.

52. Saini I, Singh D, Khosla A (2013) QRS detection using K-Nearest Neighbor algorithm (KNN) and evaluation on standard ECG databases. J Adv Res 4:331–344.

53. Yazdani A, Hoffmann U, Ebrahimi T (2009) Classification of EEG Signals Using Dempster Shafer Theory and a K-Nearest Neighbor Classifier. 4th Int IEEE EMBS Conf Neural Eng 327–330.

54. Cortes C, Vapnik V (1995) Support-Vector Networks. Mach Learn 20:273–297.

55. Vapnik VN (2000) The Nature of Statistical Learning Theory. Springer New York, New York, NY

56. Roobaert D (2000) {DirectSVM: a fast and simple support vector machine perceptron}. Neural Netw Signal Process X 2000 Proc 2000 IEEE Signal Process Soc Workshop.

57. Chen F, Tang B, Song T, Li L (2014) Multi-fault diagnosis study on roller bearing based on multi-kernel support vector machine with chaotic particle swarm optimization. Measurement 47:576–590.

58. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133.

59. Hornik K, Stinchcombe M, White H (1989) Multilayer Feedforward Networks Are Universal Approximators. Neural Netw 2:359–366.

60. Yu H, Xie T, Paszczynski S, Wilamowski BM (2011) Advantages of Radial Basis Function Networks for Dynamic System Design. IEEE Trans Ind Electron 58:5438–5450.

61. Lee C-C, Chung P-C, Tsai J-R, Chang C-I (1999) Robust radial basis function neural networks. IEEE Trans Syst Man Cybern Part B Cybern 29:674–685.

62. Jain AK, Mao J, Mohiuddin KM (1996) Artificial neural networks: a tutorial. Computer 29:31–44.

63. Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27:861–874.

64. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143:29–36.

65. Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit 30:1145–1159.

66. Egan JP (1975) Signal Detection Theory and Roc Analysis. Academic Pr, New York, NY

67. Swets JA, Dawes RM, Monahan J (2000) Better decisions through science. Sci Am 283:82–87.

68. Provost F, Fawcett T (2001) Robust Classification for Imprecise Environments. Mach Learn 42:203–231.

69. Casillas J, Cordón O, Herrera F, Magdalena L (2003) Interpretability Improvements to Find the Balance Interpretability-Accuracy in Fuzzy Modeling: An Overview. In: Casillas DJ, Cordón DO, Herrera DF, Magdalena DL (eds) Interpret. Issues Fuzzy Model. Springer Berlin Heidelberg, pp 3–22

70. Zadeh LA (1973) Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Trans Syst Man Cybern SMC-3:28–44.

71. Casillas J, Cordón O, Herrera F, Magdalena L (2003) Accuracy Improvements in Linguistic Fuzzy Modeling. Springer Berlin Heidelberg, Berlin, Heidelberg

72. Casillas J, Cordón O, Herrera F, Magdalena L (2003) Interpretability Issues in Fuzzy Modeling. Springer Berlin Heidelberg, Berlin, Heidelberg

73. Saez JA, Luengo J, Herrera F (2011) Fuzzy Rule Based Classification Systems versus crisp robust learners trained in presence of class noise's effects: A case of study. In: 2011 11th Int. Conf. Intell. Syst. Des. Appl. ISDA. pp 1229–1234

74. Gacto MJ, Alcalá R, Herrera F (2011) Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. Inf Sci 181:4340–4360.

75. Al-Jarrah OY, Yoo PD, Muhaidat S, et al (2015) Efficient Machine Learning for Big Data: A Review. Big Data Res 2:87–93.

76. Zhongdong W, Jianping Y, Weixin X, Xinbo G (2004) Reduction of training datasets via fuzzy entropy for support vector machines. In: 2004 IEEE Int. Conf. Syst. Man Cybern. IEEE Cat No04CH37583. pp 2381–2385 vol.3

77. Vishwanathan SVN, Murty MN (2002) Use of Multi-category Proximal SVM for Data Set Reduction. In: Abraham PA, Köppen M (eds) Hybrid Inf. Syst. Physica-Verlag HD, pp 19–24

78. Ding C, He X (2004) K-means Clustering via Principal Component Analysis. In: Proc. Twenty-First Int. Conf. Mach. Learn. ACM, New York, NY, USA, pp 225–232

79. Wang S, Li Z, Liu C, et al (2014) Training data reduction to speed up SVM training. Appl Intell 41:405–420.

80. Wang Z, Zhu C, Niu Z, et al (2014) Multi-kernel classification machine with reduced complexity. Knowl-Based Syst 65:83–95.

81. Caruana R, Niculescu-Mizil A (2006) An Empirical Comparison of Supervised Learning Algorithms. In: Proc. 23rd Int. Conf. Mach. Learn. ACM, New York, NY, USA, pp 161–168

82. Kotsiantis SB (2007) Supervised Machine Learning: A Review of Classification Techniques. In: Proc. 2007 Conf. Emerg. Artif. Intell. Appl. Comput. Eng. Real Word AI Syst. Appl. EHealth HCI Inf. Retr. Pervasive Technol. IOS Press, Amsterdam, The Netherlands, The Netherlands, pp 3–24

83. E. Jeong KF Identification of Transient in Nuclear Power Plant Using Adaptive Template Matching with Neural Network.

84. Mol AC de A, Martinez AS, Schirru R (2003) A neural model for transient identification in dynamic processes with “don’t know” response. Ann Nucl Energy 30:1365–1381.

85. Zhang J, He ZY, Lin S, et al (2013) An ANFIS-based fault classification approach in power distribution system. Int J Electr Power Energy Syst 49:243–252.

86. Dogantekin E, Avci E, Erkus O (2013) Automatic RNA virus classification using the Entropy-ANFIS method. Digit Signal Process 23:1209–1215.

87. Barakat S, Eteiba MB, Wahba WI (2014) Fault location in underground cables using ANFIS nets and discrete wavelet transform. J Electr Syst Inf Technol 1:198–211.

88. Son LH, Linh ND, Long HV (2014) A lossless DEM compression for fast retrieval method using fuzzy clustering and MANFIS neural network. Eng Appl Artif Intell 29:33–42.

89. (2002) Principal Component Analysis. Springer-Verlag, New York

90. Wang H (2012) Block Principal Component Analysis with L1-norm for Image Analysis. Pattern Recogn Lett 33:537–542.

91. Maier HR, Dandy GC (2000) Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environ Model Softw 15:101–124.

92. Bankó Z, Abonyi J (2012) Correlation based dynamic time warping of multivariate time series. Expert Syst Appl 39:12814–12823.

93. Karamitopoulos L, Evangelidis G, Dervos D (2010) PCA-based Time Series Similarity Search. In: Stahlbock R, Crone SF, Lessmann S (eds) Data Min. Springer US, pp 255–276

94. Li H (2014) Asynchronism-based principal component analysis for time series data mining. Expert Syst Appl 41:2842–2850.

95. Segreto T, Simeone A, Teti R (2014) Principal component analysis for feature extraction and NN pattern recognition in sensor monitoring of chip form during turning. CIRP J Manuf Sci Technol 7:202–209.

96. Jang J-SR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23:665–685.

97. Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern SMC-15:116–132.

98. Cheng C-B, Lee ES (1999) Applying fuzzy adaptive network to fuzzy regression analysis. Comput Math Appl 38:123–140.

99. Cheng C-B, Cheng C-J, Lee ES (2002) Neuro-fuzzy and genetic algorithm in multiple response optimization. Comput Math Appl 44:1503–1514.

100. Costa S, Fiori S (2001) Image compression using principal component neural networks. Image Vis Comput 19:649–668.

101. Wang Z, Leung C-S, Wong T-T, Zhu Y-S (2004) Eigen-image based compression for the image-based relighting with cascade recursive least squared networks. Pattern Recognit 37:1219–1231.

102. Wang Z, Leung C-S, Zhu Y-S, Wong T-T (2004) Data compression on the illumination adjustable images by PCA and ICA. Signal Process Image Commun 19:939–954.

103. Cattell RB (1966) The Scree Test For The Number Of Factors. Multivar Behav Res 1:245–276.

104. D'agostino RB, Russell HK (2005) Scree Test. Encycl. Biostat.

105. Zhu M, Ghodsi A (2006) Automatic dimensionality selection from the scree plot via the use of profile likelihood. Comput Stat Data Anal 51:918–930.

106. Japkowicz N, Stephen S (2002) The Class Imbalance Problem: A Systematic Study. Intell Data Anal 6:429–449.

107. Japkowicz N (2001) Concept-Learning in the Presence of Between-Class and Within-Class Imbalances. In: Stroulia E, Matwin S (eds) Adv. Artif. Intell. Springer Berlin Heidelberg, pp 67–77

108. Nahar J, Imam T, Tickle KS, et al (2012) Computational intelligence for microarray data and biomedical image analysis for the early diagnosis of breast cancer. Expert Syst Appl 39:12371–12377.

109. Sun T, Zhang R, Wang J, et al (2013) Computer-Aided Diagnosis for Early-Stage Lung Cancer Based on Longitudinal and Balanced Data. PLoS ONE 8:e63559.

110. Dal Pozzolo A, Caelen O, Le Borgne Y-A, et al (2014) Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst Appl 41:4915–4928.

111. Fawcett T, Provost F (1997) Adaptive Fraud Detection. Data Min Knowl Discov 1:291–316.

112. Chairi I, Alaoui S, Lyhyaoui A (2012) Intrusion Detection based Sample Selection for imbalanced data distribution. In: 2012 Second Int. Conf. Innov. Comput. Technol. INTECH. pp 259–264

113. Cieslak DA, Chawla NV, Striegel A (2006) Combating imbalance in network intrusion datasets. In: 2006 IEEE Int. Conf. Granul. Comput. pp 732–737

114. Triguero I, del Río S, López V, et al (2015) ROSEFW-RF: The winner algorithm for the ECBDL'14 big data competition: An extremely imbalanced big data bioinformatics problem. Knowl-Based Syst 87:69–79.

115. Peng L, Zhang H, Chen Y, Yang B Imbalanced traffic identification using an imbalanced data gravitation-based classification model. Comput Commun.

116. Brown I, Mues C (2012) An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst Appl 39:3446–3453.

117. Chawla NV (2005) Data Mining for Imbalanced Datasets: An Overview. In: Maimon O, Rokach L (eds) Data Min. Knowl. Discov. Handb. Springer US, pp 853–867

118. He H, Garcia EA (2009) Learning from Imbalanced Data. IEEE Trans Knowl Data Eng 21:1263–1284.

119. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic Minority Over-sampling Technique. J Artif Int Res 16:321–357.

120. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2009) Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. In: Theeramunkong T, Kijsirikul B, Cercone N, Ho T-B (eds) Adv. Knowl. Discov. Data Min. Springer Berlin Heidelberg, pp 475–482

121. Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In: Lavrač N, Gamberger D, Todorovski L, Blockeel H (eds) Knowl. Discov. Databases PKDD 2003. Springer Berlin Heidelberg, pp 107–119

122. Gao M, Hong X, Chen S, Harris CJ (2011) A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems. Neurocomputing 74:3456–3466.

123. Han H, Wang W-Y, Mao B-H (2005) Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang D-S, Zhang X-P, Huang G-B (eds) Adv. Intell. Comput. Springer Berlin Heidelberg, pp 878–887

124. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: IEEE Int. Jt. Conf. Neural Netw. 2008 IJCNN 2008 IEEE World Congr. Comput. Intell. pp 1322–1328

125. Li J, Li H, Yu J-L (2011) Application of Random-SMOTE on Imbalanced Data Mining. In: 2011 Fourth Int. Conf. Bus. Intell. Financ. Eng. BIFE. pp 130–133

126. Zeng Z-Q, Gao J (2009) Improving SVM Classification with Imbalance Data Set. In: Leung CS, Lee M, Chan JH (eds) Neural Inf. Process. Springer Berlin Heidelberg, pp 389–398

127. Zhai Y, Ma N, Ruan D, An B (2011) An effective over-sampling method for imbalanced data sets classification. Chin J Electron 20:489–494.

128. Estabrooks A, Jo T, Japkowicz N (2004) A Multiple Resampling Method for Learning from Imbalanced Data Sets. Comput Intell 20:18–36.

129. Milord JT, Perry RP (1977) A Methodological Study of Overloadx. J Gen Psychol 97:131–137.

130. Cohen S (1980) Aftereffects of stress on human performance and social behavior: a review of research and theory. Psychol Bull 88:82–108.

131. Speier C, Valacich JS, Vessey I (1999) The influence of task interruption on individual decision making. Decis. Sci. 30: