Meta-cognitive learning algorithm for neuro-fuzzy inference systems

Kartick Subramanian

2014

Kartick Subramanian. (2014). Meta-cognitive learning algorithm for neuro-fuzzy inference systems. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61828>

<https://doi.org/10.32657/10356/61828>

Meta-cognitive Learning Algorithm for Neuro-Fuzzy Inference Systems

Kartick Subramanian

School of Computer Engineering

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

2014

My prayers to God almighty.

Dedicated to my parents, who have been there with me throughout.

Acknowledgement

Firstly I would like to thank my supervisor, Dr. Suresh Sundaram. Words wont be enough to express my gratitude towards him for his constant support, and push at times. This thesis would not have even conceptualized without his help. Next I would like to thank Dr. N. Sundararajan for the discussions during the my Ph.D.

Next, I would like to thank all my family and friends for providing me with an amiable platform to work and play. Their support and criticism have helped a lot during my research.

Finally, I would like to thank Center for Computational Intelligence, School of Computer Engineering and Nanyang Technological University for providing me with such an infrastructure and support to conduct my research.

Contents

Acknowledgements	i
Abstract	viii
List of Figures	xii
List of Tables	xv
List of Symbols	xviii
List of Abbreviations	xxi
1 Introduction	1
1.1 Background and Motivation	1
1.2 Issues and Challenges	6
1.3 Research Objectives	10
1.4 Research Contributions	12
1.5 Thesis Organization	14
2 Literature Review	19
2.1 Introduction	19
2.2 Fuzzy Inference Mechanisms	23
2.2.1 Mamdani Fuzzy Inference Mechanism	23
2.2.2 Takagi-Sugeno-Kang Fuzzy Inference Mechanism	24
2.2.3 Other Inference Mechanisms	26
2.3 Membership Functions	27
2.3.1 Non-Differentiable Membership Functions	27
2.3.2 Differentiable Membership Functions	28
2.4 Learning Modes	30
2.4.1 Batch/ Offline Learning Mode	30

2.4.2	Sequential/ Online Learning Mode	33
2.5	Applications	38
2.5.1	System Control	38
2.5.2	Soft Computing	39
2.6	Summary	41

3 A Meta-Cognitive Sequential Learning Algorithm for Neuro-Fuzzy Inference System (McFIS) 43

3.1	Introduction	43
3.2	Meta-Cognitive Neuro-Fuzzy Inference System	46
3.2.1	Problem Definition	47
3.2.2	Cognitive Component: The Neuro-Fuzzy Inference System	48
3.2.3	Meta-cognitive Component: The Self-Regulatory Learning Mechanism	49
3.2.4	Discussion on the Influence of Self-Regulatory Thresholds	57
3.3	Performance Evaluation of McFIS for Function Approximation Problems	58
3.3.1	Performance measures	59
3.3.2	Non-linear System Identification Problem 1	59
3.3.3	System Identification Problem 2	63
3.3.4	Mackey-Glass Time Series Problem	65
3.3.5	Box-Jenkins Gas Furnace Problem	66
3.4	Discussion on Effects of Meta-Cognition	67
3.5	Performance Evaluation of McFIS for Classification Problems	68
3.5.1	Performance measures	68
3.5.2	Performance comparison	69
3.6	Summary	70

4 A Meta-Cognitive Sequential Learning Algorithm for Neuro-Fuzzy Inference System based Classifier (McFIS-Classifier) 73

4.1	Introduction	73
4.2	Meta-Cognitive Neuro-Fuzzy Inference System based Classifier	75
4.2.1	Problem Definition	75

4.2.2	Cognitive Component: The Neuro-Fuzzy Inference System	76
4.2.3	Meta-Cognitive Component: The Self-Regulatory Learning Mechanism	78
4.2.4	Discussion on the Influence of Self-Regulatory Thresholds	86
4.3	Performance Evaluation of McFIS-classifier	88
4.3.1	Dataset Description	88
4.3.2	Performance Measures	90
4.3.3	Performance Comparison	91
4.3.4	Statistical Comparison	94
4.4	Discussion on the Effect of Meta-Cognition	95
4.5	Summary	97
5	Meta-Cognitive Neuro-Fuzzy Inference System based Classifier for Human Action Recognition	99
5.1	Overview	99
5.2	Human Action Recognition based on 2D Video Sequence	101
5.2.1	Motivation	101
5.2.2	Background and Related Works on 2D Action Recognition	102
5.2.3	Action Representation	104
5.2.4	Performance Evaluation of 2D Action Recognition System	108
5.3	Human Action Recognition based on 3D Video Sequence	114
5.3.1	Motivation	114
5.3.2	Background and Related Work on 3D Action Recognition	115
5.3.3	3D Optical Flow Estimation	116
5.3.4	Performance Evaluation of 3D Action Recognition System	117
5.4	Summary	119
6	Meta-Cognitive Neuro-Fuzzy Inference System based Classifier for Facial Emotion Recognition	121
6.1	Overview	122
6.2	Motivation	123
6.3	Review of Major Works in Facial Emotion Recognition	125

6.4	Feature Extraction Procedure	127
6.4.1	Pixel based Feature Extraction	127
6.4.2	Local Binary Pattern based Feature Extraction	127
6.4.3	Scale Invariant Feature Transform based Feature Extraction	129
6.4.4	Gabor Filter based Feature Extraction	130
6.4.5	Curvelet based Feature Extraction	130
6.5	Performance Study	132
6.5.1	Dataset Description	133
6.5.2	10-Fold Cross Validation Study	134
6.5.3	Person Independent Study	135
6.5.4	Database Independent Study	137
6.6	Summary	138
7	Conclusions and Future Direction	141
7.1	Thesis Summary	141
7.1.1	Self-Regulatory Learning Mechanism	142
7.1.2	McFIS	143
7.1.3	McFIS-classifier	144
7.1.4	McFIS-classifier for human action recognition	145
7.1.5	McFIS-classifier for human emotion recognition	146
7.2	Future Directions	146
7.2.1	Improvements from Meta-Cognitive Principles	146
7.2.2	Improvements in Fuzzy Inference Mechanism	148
7.2.3	Improvements in Neural Learning Algorithms	150
7.2.4	Issues Related to Input and Output Features	151
	List of Author's Publications	153
	Bibliography	156

Abstract

Neuro-fuzzy systems are learning machines that employ algorithms derived from artificial neural networks to find the parameters of a fuzzy inference system. These hybrid-intelligent systems can learn fuzzy rules from the data, while preserving their semantic properties. The learning algorithms employed in these systems are inspired from the physiology and functioning of human brain. As the study on neuroscience and cognitive psychology is an ongoing process, it is essential that these machine learning algorithms be extended with developments from the findings in human learning psychology.

Studies in human learning psychology have shown that self-regulated learning in a meta-cognitive framework is the best learning strategy. Meta-cognition also provides a learner with self-monitoring, which is a step-by-step process of evaluation during the learning process. In this thesis, we develop a Meta-Cognitive sequential learning algorithm for Neuro-Fuzzy Inference System called McFIS which is based on a well-known model of meta-cognition proposed by Nelson and Narens. Similar to Nelson and Narens model of meta-cognition, McFIS consists of a cognitive and a meta-cognitive component. A Takagi-Sugeno-Kang type neuro-fuzzy inference system forms the cognitive component and a self-regulatory learning mechanism is its meta-cognitive component. The meta-cognitive component monitors the knowledge in the cognitive component and controls the learning by efficiently deciding on which sample to learn, when to learn it and how to learn it, efficiently. Such a learning strategy helps the network generalize the functional relationship between input and output. The performance evaluation of McFIS on a set of benchmark function approximation and time-series prediction problems indicate significant improvement over other state-of-the-art techniques. In addition, quantitative study on a set classification problems indicate motivating results.

It has been shown in literature that algorithms developed to solve function approximation problems may not solve classification problems efficiently. For solving classification problems, class-specific information plays a vital role. Hence, McFIS has been extended to solve classification problems, efficiently, by considering class-specific information. Here, the meta-cognitive component monitors the class-specific knowledge in the cognitive component and decides on efficient control strategy. Moreover, the learning algorithm also considers class-specific information while learning the fuzzy rules. This helps the network learn the classification decision boundary efficiently. The performance of McFIS based classifier is studied on a set of binary and multi-category classification problems. The performance comparison indicates superior classification ability of McFIS based classifier over other classifiers considered in literature.

The proposed McFIS based classifier is used to solve two important problems in video analytics: action recognition and emotion recognition. The problem of human action recognition and emotion recognition are immensely difficult due to lack of features to accurately represent various actions/emotions, and lack of efficient classifiers. In the literature of action recognition, it has been shown that optical flow based features can capture apparent movement in object naturally. Moreover, they can represent information hierarchically from local pixel level to global object level. Hence, we develop McFIS based action recognition system, where the aim of McFIS is to find the optimal decision boundary separating different actions, based on their optical flow based features. We then propose a 3D action recognition system. Here, 3D optical flow features, which are combination of 2D optical flow based features with depth flow features, are employed. The use of 3D video sequence help in avoiding issues such as occlusion and view point variation. The performance of McFIS based action recognition system is evaluated on a set of benchmark as well as proprietary action recognition data sets. The performance analysis against other features and techniques indicate superior performance of McFIS with optical flow features.

Further, we extend McFIS for human emotion recognition problems. From two benchmark datasets, we have extracted pixel based, Gabor filter, local binary pattern, curvelet and scale invariant feature transform based features, in order to study their effect on emotion recognition. Based on the extracted data, we perform a stratified 10-fold cross-validation study. We also conduct person independent and data set independent studies

to validate effectiveness of the proposed emotion recognition system. The statistical performance study clearly indicates the emotion recognition ability of McFIS based classifier over other results reported in the literature.

In the future, meta-cognitive learning mechanism will be extended to Type-2 neuro-fuzzy inference systems. Moreover, use of other models of meta-cognition will also be explored.

List of Figures

1.1	Basic Structure of Human Cognitive Processes as Proposed by Nelson and Narens [145]	5
1.2	Thesis Organization	15
2.1	Fusion Level Categorization of Artificial Neural Networks and Fuzzy Inference Systems Adapted from [214]	20
3.1	(a) Nelson and Narens Model of Meta-Cognition Adapted from [145] and (b) its Analogy in McFIS	44
3.2	Architecture of Neuro-Fuzzy Inference System. The Inset Figure Describes the internal Structure of the i^{th} Gaussian Node.	48
3.3	Schematic Diagram Highlighting the Self-regulating Learning Mechanism in McFIS.	50
3.4	Training Logarithmic Prediction Error for Non-Linear System Identification Problem 1 on McFIS	60
3.5	Illustration of Sample Deletion Strategy of McFIS for Non-Linear System Identification Problem 1	61
3.6	Depiction of Variation in Spherical Potential, Prediction Error, and Self-Regulatory Thresholds of McFIS for System Identification Problem 1 . .	62
4.1	Nelson and Narens Model of Meta-Cognition Adapted from [145] and its Analogy in McFIS-Classifier	75
4.2	Architecture of Neuro-fuzzy Inference System. The inset Figure Describes the Internal Structure of the i^{th} Gaussian Node.	77
5.1	Overview of McFIS based Action Recognition System	100

5.2	Example of Optical Flow Feature Extraction Process with Weizmann Data Set	105
5.3	Optical Flow based Feature Extraction Methodology as Employed in this Thesis	107
5.4	Illustration of Actions Performed in Weizmann Data Set [68]	109
5.5	Illustration of Actions Performed in KTH Data Set [184]	109
5.6	A Snapshot of Sample 2D (Bend, Swim and Kick) Actions (a) - (c) and the Corresponding Depth Images (d) - (f) Used in 3D Action Recognition	118
6.1	Overview of McFIS Based Emotion Recognition System	122
6.2	Image Cropping Technique Employed for Emotion Recognition Study . .	128
6.3	Example for Local Binary Pattern Feature Extraction Process for Emotion Recognition Using Image TM.NE2.178 from JAFFE Database [135] . . .	128
6.4	Scale Invariant Feature Transform Feature Extraction Process for Emotion Recognition Using Image TM.NE2.178 from JAFFE Database [135] . . .	129
6.5	The Non-Overlapping Patches in Image TM.NE2.178 from JAFFE Database [135] for Extraction of Gabor Filter Based Features	131
6.6	Scale and Angle Representation of Curvelet Transform Used in Feature Extraction for Emotion Recognition	131
6.7	Curvelet Features Extracted from an image TM.NE2.178 from JAFFE database [135] for Emotion Recognition	132
6.8	Sample Facial Images from JAFFE [135] Database for Emotion Recognition	133
7.1	Theoretical Framework of Monitoring and Control of Human Meta-Memory adapted from [145]	147

List of Tables

2.1	Summary of Major Types of Algorithms Discussed in Literature Review .	42
3.1	Performance Comparison of McFIS with Other State-of-the-art Methods for Non-linear System Identification Problem 1	64
3.2	Performance Comparison of McFIS with Other State-of-the-art Methods for Non-linear System Identification Problem 2	64
3.3	Performance Comparison of McFIS with Other State-of-the-art Methods for Mackey-Glass Stochastic Time Series Problem	65
3.4	Performance Comparison of McFIS with Other State-of-the-art Methods for Box-Jenkins Gas Furnace Problem	66
3.5	Quantitative Verification of McFIS Learning Philosophy in SAFIS Algo- rithm [169] Using System Identification Problem 1	68
3.6	Description of Benchmark Data Sets Employed in Chapter 3	69
3.7	Performance Comparison of McFIS with Other State-of-the-art Methods on Classification Problems	70
4.1	Dataset Specifications for Classification using McFIS-Classifer	89
4.2	Performance Evaluation of McFIS-classifier with Other State-of-the-art Methods on Binary Classification Problems	92
4.3	Performance Evaluation of McFIS-classifier with Other State-of-the-art Methods on Multi-class classification problems	93
4.4	Quantitative Verification of McFIS Learning Philosophy in SAFIS Algo- rithm [169] using Classification Problems	96
5.1	Performance Comparison of McFIS-classifier with SVM on Frame Level Weizmann Data Set	110

5.2	Performance Comparison of McFIS-classifier with SVM on Frame Level KTH Data Set	111
5.3	Testing Confusion Matrix of Weizmann Sub-Sequence Level Features Dataset with McFIS-classifier	112
5.4	Testing Confusion Matrix of KTH Sub-Sequence Level Features Dataset with McFIS-classifier	112
5.5	Performance Comparison of McFIS-classifier with SVM on Sub-Sequence Level Weizmann Data Set	112
5.6	Performance Comparison of McFIS-classifier with SVM on Sub-Sequence Level KTH Data Set	113
5.7	Performance Comparison of McFIS-classifier with State-of-the-art Action Recognition System for Weizmann Data Set	114
5.8	Performance Comparison of McFIS-classifier with State-of-the-art Action Recognition System for KTH Data Set	114
5.9	Performance Comparison of McFIS-classifier based 3D Action Recognition System with SVM on 10-Fold Cross Validation Data Set	119
6.1	Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM on Five Features extracted from JAFFE 10-Fold Cross Validation Database	134
6.2	Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM on Five Features extracted from TFEID 10-Fold Cross Validation Database	134
6.3	Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM on Five Features extracted from JAFFE Person Independent Database	136
6.4	Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM on Five Features extracted from TFEID Person Independent Database	136
6.5	Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM Trained on Five Features extracted from JAFFE Database and Tested on TFEID	137

6.6	Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM Trained on Five Features extracted from TFEID Database and Tested on JAFFE	138
-----	---	-----

List of Symbols

α		Element of output weight of network
β		Contribution of a rule to the total output
γ		Directional Gaussian blurring factors
δ		Slope of control parameters for self-adaptivity
η		Inter-class overlap factor
η_o		Overall classification accuracy
η_a		Average classification accuracy
η_g		Geometric classification accuracy
θ		Measure of angle
κ		Intra-class overlap factor
μ		Center of a fuzzy rule
σ		Width of a fuzzy rule
ϕ		Membership of a fuzzy rule
ψ		Spherical potential
ν		Lag to a non-linear system
χ		Lag to a non-linear system
$\mathbb{R}$		Set of real numbers
$\mathbf{x}$		Input feature vector
m		Number of input features
n		Number of output features
$\mathbf{w}$		Network parameters
u		input to a non-linear system
nr		Nearest rule
T		Transpose

$\mathbf{e}$	Prediction error
E	Prediction error
E_d	Sample delete threshold
E_a	Self-adaptive sample addition threshold
E_l	Self-adaptive sample update threshold
E_p	Rule pruning threshold
E_w	Rule pruning window
E_S	Novelty threshold
PS	Percentage of samples used for training
p_0	Initial error covariance
q_0	Initial process noise
r_0	Initial measurement noise
$\mathbf{R}$	measurement noise Matrix
$\mathbf{G}$	Kalman gain matrix
$\mathbf{P}$	Error covariance matrix
$\mathbf{a}$	Gradient of output with respect to parameters
N	Total number of training samples
p	Posterior probability
c	Class label
$\hat{}$	Predicted value of a given function or variable
I	Identity matrix
G	Accumulated optical flow

List of Abbreviations

NFIS	Neuro-Fuzzy Inference Systems
McFIS	Meta-Cognitive Neuro-Fuzzy Inference System
ANOVA	Analysis of Variance
TSK	Takagi-Sugeno-Kang
T-norm	Triangular Norm
S-norm	A.K.A T-conorm, Triangular Co-Norm
FALCON	Fuzzy Logic Control
ANFIS	Adaptive Network-based Fuzzy Inference System
NEFCON	Neuro-Fuzzy Control
NEFPROX	Neuro-Fuzzy Approximator
NEFCLASS	Neuro-Fuzzy Classifier
eTS	Evolving Takagi-Sugeno Model
simpl_eTS	Simplified Evolving Takagi Sugeno Model
SAFIS	Sequential Adaptive Fuzzy Inference System
SOFMLS	Self-Organizing Fuzzy Modified Least Squares
ENFM	Evolving Neuro-Fuzzy Model
ELM	Extreme Learning Machine
OS-Fuzzy-ELM	Online Sequential Fuzzy Extreme Learning Machine
eClass	Evolving Classifier
eClass0	Evolving Type-0 Classifier
eClass1	Evolving Type-1 Classifier
SOFNN	Self Organizing Fuzzy Neural Network
SVM	Support Vector Machine
KNN	K Nearest Neighbour

UCI	University of California, Irvine
EEG	Electroencephalography
RMSE	Root Mean Squared Error
PS	Percentage of Samples Employed
NDEI	Non Destructive Error Index
S.D	Standard Deviation
IS	Image Segmentation
GI	Glass Identification
LD	Liver Disorder
ION	Ionosphere
BC	Breast Cancer
AE	Acoustic Emission
VC	Vehicle Classification
Sat	Satellite Imaging
GCM	Global Cancer Map
I.F.	Imbalance Factor
2D	Two Dimensional
3D	Three Dimensional
HOG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
LBP	Local Binary Pattern
LoG	Laplacian of Gaussian
JAFFE	Japanese Female Facial Expression
TFEID	Taiwanese Facial Expression Identification

Chapter 1

Introduction

1.1 Background and Motivation

The brain is one of the most complicated organs present in the human body consisting of various interconnected components associated with specific functionalities. Many research works are being conducted in the field of medicine, to clearly understand the task of each of these components [133, 127]. Apart from the above, researches are also being conducted in the field of psychology, to understand how different parts of brain affects the unique human traits of higher order thinking, consciousness, reasoning, emotion, language, etc. In many aspects, scientists have been successful in understanding the working of human brain [127, 22, 240].

In artificial intelligence community, researchers aim at emulating human-like learning and thinking in machines, by modeling different aspects of human brain. Various brain inspired computing models have been proposed, including Hebbian network [76], Hopfield network [77], which address the memory aspect of brain, perceptron [136], radial basis function network [140], based on the information processing ability of human brain, cerebellar model articulation controller [6], based on cerebral cortex and self organizing maps [107] which is based on visual cortex.

It is well known that these artificial neural networks can accurately model the functional relationship between the input and the output, due to the connectionism and parallelism derived from the brain. Further, these networks have also been bestowed with findings from fuzzy set theory [254] to enable them with imprecise knowledge representation and human-like reasoning ability. Few such networks include fuzzy neural networks

[94, 98] which combines neural networks based learning in fuzzy inference systems, fuzzy cognitive maps [109] which mimics human reasoning ability, fuzzy cerebellar model articulation controller [82] where cerebellar model articulation controller is manipulated using fuzzy set theory.

Neuro-fuzzy inference systems are hybrid intelligence systems which synergies human-like reasoning in fuzzy system with the learning and connectionist structure in neural networks. This combination helps the fuzzy inference system achieve twin requirements of accurate knowledge acquisition and knowledge interpretability/representability. As a result, these neuro-fuzzy inference systems have found a wide range of applications. Hence, in this thesis such a neuro-fuzzy inference system is employed to efficiently model a given input output relationship.

Over the past few decades, researches have been conducted to enhance the above mentioned networks with various aspects of human pedagogy. Supervised learning, such as feedforward [76], feedback [77], competitive [33], reinforcement learning [212], has sought to teach the networks the functional relationship between the input and output for cognitive tasks like recognition, representation and reproduction. Unsupervised learning, such as competitive learning [107], dimension reduction [76], auto-association [80], has aimed to teach the networks to extract the underlying knowledge from the data as well as to compress it.

The learning methodology and learning techniques mentioned above have helped the networks mimic the primary functionalities of different cognitive processes. However, very few works in machine learning literature consider the human psychological aspect of learning. In humans, it has been shown that self-regulation in learning is very important and influences the learning ability of a person significantly [29, 195]. It provides humans with the ability of higher order thinking, conscious reasoning and active control over learning. Literatures on human learning indicate that meta-cognition refers to higher order thinking involved in cognitive tasks such as planning, monitoring, and evaluating progress towards completion of a task. It provides humans with self-regulatory mechanism, which results in best learning strategy, where one can assess their current knowledge, identify when a new knowledge is needed, as well as provide strategies to acquire that new knowledge [90, 166, 239].

The term “meta-cognition” was first coined by John Flavell [56] and defined it as *ones knowledge concerning their own cognitive processes and products or anything related to them*. Meta-cognition consists of two components: meta-cognitive knowledge and meta-cognitive regulation. Meta-cognitive regulation involves use of two processes: meta-cognitive monitoring and meta-cognitive control [58, 57]. These terms are defined as follows:

- *Meta-cognitive knowledge* refers to acquired declarative knowledge about cognitive processes. It is composed of facts, beliefs that can be stated and employed to control the cognitive processes.
- *Meta-cognitive monitoring* refers to monitoring one’s own thought processes and one’s existing state of knowledge. It helps one plan subsequent actions for efficient learning.
- *Meta-cognitive control* refers to regulating one’s ongoing cognitive processes. It determines the factors such as planning and allocation of study time, use of learning strategies, etc.

Many analytical models relating to key concepts in meta-cognition and its influence in cognition have been reported in literature, a review of which is available in [40]. According to Wellman, [238] human meta-cognition involves separate but inter-related cognitive processes, which emerges from the awareness of the difference between the internal and external worlds. This model considers the knowledge to be classified as person variable, task variable, and strategy variable. It includes a self-monitoring component employing which one can evaluate their own level of comprehension and mental performance. Inspired by the works of Flavell and Wellman, various research works and model of meta-cognition have been recently proposed. One such work includes Perlovsky’s knowledge instinct theory [155], according to which every human have craving to be creative and this conscious intention to be creative stems from knowledge of one’s self and the environment. Nelson and Narens [145] presented a general framework for integrating meta-cognition and meta-memory, which outlines the interactive process of monitoring and control between cognition and meta-cognition. In contrast to [59], which focuses

on knowledge of memory, Nelson and Narens model describes the regulatory processes involved in meta-cognition.

Nelson and Narens' model [145] is also in line with the study on human brain analysis. Based on the human brain analysis, it has been reported that there exists a dual memory system (so called episodic and semantic memory), which could be linked to the cognitive and meta-cognitive aspects of human beings. From these studies, it could be understood that episodic memory represents the knowledge based on experience (knowledge) and semantic memory represents the knowledge extracted from experience with or without referencing to them (meta-knowledge). These two memories interact very closely (the meta-cognitive process updates its meta-knowledge through experience and controls the learning process, i.e., cognition) and effectively controls knowledge acquisition, retention and retrieval [225, 218]. Nelson and Narens model splits up the cognitive processes into two or more specifically inter-related components, referred to as meta-cognitive component and cognitive components, where the meta-cognitive component consists of a dynamic model of the cognitive component. Between meta-cognition and cognition, there are two main relationships, "monitoring" and "control", which are defined in terms of direction of flow of signals between the meta-cognitive and the cognitive components. This relationship is pictorially represented in Fig. 1.1. The signal flowing from meta-cognitive component to cognitive component is "control". The basic notion underlying control is that the meta-cognitive component modifies the cognitive component based on these signals. In particular, the information flowing from the meta-cognitive component changes either the state of the cognitive component or the cognitive component itself. Since the control signal in itself does not yield any information about the cognitive component, a logically independent "monitory" signal is employed from cognitive component to meta-cognitive component. The objective of this signal is to inform the meta-cognitive component about the state of the cognitive component, based on which the meta-cognitive component changes its knowledge about the cognitive component. Furthermore, in this model, the distinction between monitoring, control, and the information processing perspective is highly compatible with the views presented in the computational literature [40].

In the field of artificial neural networks, a few algorithms have been developed which address the meta-cognitive aspect of learning [211, 177] and it has been extended to

Figure 1.1: Basic Structure of Human Cognitive Processes as Proposed by Nelson and Narens [145]

complex-valued neural networks [179, 180]. Current state-of-the-art in neuro-fuzzy literature addresses pure cognitive aspect of human learning inspired by human brain. A detailed review on neuro-fuzzy inference systems and their learning algorithms and modes is provided in chapter 2. Very few research works in neuro-fuzzy literature address the psychological aspect of human learning. There is a need to model the cognitive processes and their inter-relationship with the self-regulatory aspects in a neuro-fuzzy framework for efficient learning. In many real-world applications such as in video analytics, the samples to train the network are not available *a priori*. As a result, in such problems, sample selection has to be conducted along with learning. In an online sequential machine learning framework, there is a need to combine sample selection and choice of learning strategies to achieve better generalization ability. This thesis shall deal with the study of meta-cognitive model for knowledge representation and self-regulated learning, in an online, sequential learning environment. Also, the interaction between key concepts of meta-cognition and their influence in learning performance will be studied.

The next section describes the issues and challenges faced in developing a meta-cognitive learning algorithm for a neuro-fuzzy inference system.

1.2 Issues and Challenges

This section discusses the major issues and challenges in incorporating human-like meta-cognition in neuro-fuzzy framework. In addition, issues in neuro-fuzzy inference systems and challenges in their learning algorithms are also highlighted.

- **Model of Meta-Cognition for Machine Learning:** Meta-cognition refers to higher order thinking that involves active control over the thinking processes involved in learning. Human meta-cognition, as described by Nelson and Narens [145] involves at least two inter-related components: cognitive component and meta-cognitive component. The meta-cognitive component consists of two key concepts: meta-cognitive knowledge and self-regulation. The meta-cognitive component controls the acquisition of knowledge in the cognitive component by monitoring the knowledge in the cognitive component. It controls by deciding on one of the following three strategies:

- *Termination of study:* Decision to stop studying an item currently being studied.
- *Item selection:* Decision about whether to study an item for an upcoming test.
- *Selection of kind of processing:* Selection of strategies to employ when attempting to commit an item to memory.

These three strategies help a learner effectively decide *what-to-learn*, *when-to-learn* and *how-to-learn*.

- *What-to-learn:* Requires an ability to identify the specific piece of knowledge that is lacking in the network.
- *When-to-learn:* Requires an ability to judge when to start and stop learning.
- *How-to-learn:* Requires the knowledge of available learning strategies, for a given task, and the ability to choose the best strategy to learn what is required.

Realization of Meta-cognitive knowledge acquisition strategies vary according to the task that has to be performed. In this thesis, a neuro-fuzzy inference system is employed for two tasks: regression and classification. In regression, the meta-cognitive

component should monitor and control the learning such that the underlying functional relationship between the input and the output is accurately captured using a compact network. In classification tasks, the aim is to learn the decision boundary based on class-specific information and thereby classify the input accurately.

Framework of human meta-cognition, as proposed by Nelson and Narens has three major roles: in knowledge acquisition, knowledge retention and knowledge retrieval. In order to perform any of the three roles, the meta-cognition depends on monitory and control signals. There is a need to model all the three roles of meta-cognition and their corresponding monitory and control signals to achieve superior performance.

- **Neuro-Fuzzy Inference Systems:** In this thesis, a neuro-fuzzy inference system forms the cognitive component to store knowledge and to infer the output based on the stored knowledge. Various fuzzy inference mechanisms are available for these systems, such as Mamdani inference mechanism [131], which focuses on interpretability of derived fuzzy rules, Takagi-Sugeno-Kang [215], which aims at predicting output with high degree of precision/ accuracy, combined Mamdani-Takagi-Sugeno [227], and parameter free inference [14]. In this thesis, a Takagi-Sugeno-Kang type fuzzy inference system is employed to verify the concepts of meta-cognition. These principles could be extended to fuzzy systems with other inference mechanisms.

In most of the practical applications, such as video analytics and medical analytics, the complete training data describing the input-output relationship is not available *a priori*. For these problems, classical batch learning algorithms are not ideal and instead, sequential learning algorithms are employed. This results in lesser space and time complexity for the learning algorithm. In addition, sequential learning algorithms automatically determine the minimal number of parameters and adapt them, to accurately approximate the functional relationship/ decision boundary, as described by a stream of training samples. Moreover, during rule addition, the parameters of newly added rule needs to be initialized such that prediction error is minimized. In this thesis, the twin issues of developing a sequential learning

algorithm for a neuro-fuzzy inference system and efficiently exploiting localization property of a Gaussian rule, is addressed.

- **Sequential Learning Framework:** In most of the real-world problems employing neuro-fuzzy inference systems, the data describing the problem arrive sequentially and are too enormous to be stored and be trained in offline fashion. Moreover in many applications, there is a temporal variation in the system input-output, which needs to be captured efficiently. There is also a need to develop neuro-fuzzy learning algorithms which learn the samples one-by-one and adapt the architectures depending on the data [16, 169]. In addition, since the data arrives one-at-a-time, the network must be able to select the appropriate samples for learning such that over-training can be avoided and outliers in the training data be avoided.
- **Gaussian Rule Exploitation:** In this thesis, the fuzzy inference system is modeled as realizing a Takagi-Sugeno-Kang type neuro-fuzzy inference system with a Gaussian membership function as the rule antecedent. In a sequential learning algorithm, when a new rule is added to the network, the consequent parameters of the newly added rule is initialized such that the prediction error is minimized. Usually, the consequent parameters are selected to be equal to the prediction error, for the particular sample, upon employing the existing rules. In such a formulation, the effect of the current sample has been neglected. In order to fully exploit the localization property of a Gaussian membership function, the effect of the newly added rule should also be considered while initializing the consequent parameters of a Takagi-Sugeno-Kang neuro-fuzzy inference system.
- **Classification Problems:** In this thesis, neuro-fuzzy inference systems is employed for two tasks: regression and classification. It has been shown in literature that algorithms developed to solve regression problems may not solve classification problems, efficiently. In classification problems class-specific criteria play a crucial role in determining the performance of the classifier [210]. While allocating a new rule, the rule overlap as well as the output weights should be initialized such that

misclassification is minimized. While allocating rule width, there should be minimum overlap between rules representing different classes. However, there should be significant overlap among rules representing same class. Significant overlap among rules of different classes lead to misclassification of samples, whereas, lack of overlap among rules of same class may reduce the approximation ability of the network. Hence, the width initialization is an issue in neuro-fuzzy inference system. Similarly, while initializing the output weights, the learning algorithm should penalize the inference system for letting the incorrect class get close to correct output. However, in the absence of misclassification, there should be no penalty.

- **Video Analytics:** Video analytics is the process of automatically detecting, analyzing, tracking and classifying objects in a video with the aim of identifying any interesting activities or events. The most commonly used type of video analytics include, person counting, person tracking, human action recognition, and emotion recognition. Here, the task of human action and emotion recognition are one of the end goals of many of video analytics systems for surveillance, human-computer interaction, operator fatigue detection, etc. In many of the video analytics, the data for learning is not available *a priori*, and it arrives one-by-one. Moreover, due to the enormous amount of data involved, it is not viable to store all the samples. Hence, there is a need for a sequential learning algorithm that is able to learn from the data sequentially. In this thesis, two video analytics of human action recognition and facial emotion recognition are considered.

- **Human Action Recognition:** It involves classification of various action based on features extracted from a given video sequence. In literature various feature extraction techniques have been proposed. However, optical flow based features can represent the apparent movement in objects and can represent information hierarchically from local pixel level to global object level. In this thesis, the effect of meta-cognitive learning algorithm in classifying human actions based on 2D optical flow based features is studied.

In the past decade, the use of 3D camera has increased significantly. These cameras represent objects in 3D employing depth information in combination

with corresponding 2D images. This render the images invariant to occlusion and as a result more distinguishable. There is a need to study the classification ability of actions based on such 3D video sequences. In order to extract features from these video, a 3D optical flow based feature need to be studied.

- **Facial Emotion Recognition:** In literature, various feature extraction processes have been proposed. There is a need to evaluate their ability to represent emotions and generalize them across cultures. Similarly, there is a need to evaluate the learning and generalization ability of latest machine learning classifiers while processing them online.

In literature, various feature extraction processes and learning methods have been discussed for efficiently representing and classifying human emotion based on facial expression. However, many of these approaches are not online in nature and cannot capture or generalize the decision boundary efficiently. Each of the feature extraction techniques have their own unique advantages. There is a need to study the classification ability of the proposed approach for efficiently recognizing human emotion based on facial expression.

Moreover, database independent studies such as person independent and data set independent studies have to be conducted to evaluate the generalization performance of the network in identifying different emotions based on various features.

1.3 Research Objectives

In this thesis, a generic framework of human meta-cognitive learning mechanism in a neuro-fuzzy inference system architecture is proposed. The research discusses and evaluates the inter-relationship between the meta-cognitive knowledge, and meta-cognitive regulation, such that the generalization performance of neuro-fuzzy inference system improves, significantly. The main research objectives could be summarized as follows:

- **Formulation of Self-Regulatory Learning Mechanism:** To formulate a learning mechanism wherein, the control parameters of the algorithm are adapted based

on the novelty of the training sample. The parameters of the algorithm should be adapted such that the samples with novel information content are learnt first, and samples with low information content be used to fine tune the network.

- **Development of Meta-Cognitive Neuro-Fuzzy Inference System:** Develop a meta-cognitive learning algorithm for a neuro-fuzzy inference system based on meta-cognitive framework proposed by Nelson and Narens. The learning algorithm would be employed to solve two tasks: regression and classification. The learning algorithm controls the learning by employing self-regulatory learning strategies of *what-to-learn*, *when-to-learn* and *how-to-learn*, based on the meta-cognitive knowledge in a neuro-fuzzy inference system. Aspects of meta-cognitive monitoring such as ease-of-learning, judgement-of-learn, and feeling-of-knowing need to be explored in a machine learning framework.
- **Sequential Learning Framework:** To develop a meta-cognitive learning algorithm for a neuro-fuzzy inference system in a sequential learning framework, such that the network grow, update, and as well as prune the rule, so that a parsimonious network is achieved. This network should be able to handle samples coming one-by-one and also temporally varying data.
- **Video Analytics:** Video analytics is the process of using computers to automatically detect and track object of interest with the aim of identifying any interesting activity. The most commonly used video analytics include, person counting, people tracking, action recognition, etc. In this thesis, two video analytics problems are considered: human action recognition and facial emotion recognition.
 - **Human Action Recognition:** Develop a human action recognition system for surveillance. Given a pre-processed video segment, efficient action features should be extracted, which can represent action information from local pixel level to global object level. Moreover, it should also be able to represent movement in objects, intrinsically. The developed system should be able to generalize and recognize actions, efficiently.

The developed human action recognition system should be extended to handle 3D video sequences. Representative depth features should be investigated for efficient action representation. The developed systems should be evaluated on various real-world scenarios.

- **Facial Emotion Recognition:** Develop a generic human emotion recognition system based on meta-cognitive neuro-fuzzy inference system. Further, a study should be conducted with such an emotion recognition system to understand the emotion representability of various feature representation techniques. The feature-system pair should be able to generalize emotions across different individuals, and further to different ethnicity, database. A study would be conducted to this extend to evaluate the performance of different features in generically representing emotions.

1.4 Research Contributions

The major contributions of this thesis are given below:

- **Self-Regulatory Sequential Learning Algorithm:** A self-regulatory learning algorithm which decides *what-to-learn*, *when-to-learn* and *how-to-learn* and implements one of the following strategies: *sample deletion strategy*- the current sample is deleted without learning; *sample learning strategy*- the current sample is used in the learning process by either adding/ pruning rules or by updating the parameters of existing rules; and *sample reserve strategy*- the current sample is reserved for fine tuning the parameters of the rules at a later stage.
- **Meta-Cognitive Neuro-Fuzzy Inference System:** A Meta-Cognitive sequential learning algorithm for a Neuro-Fuzzy Inference System (McFIS) has been proposed which is based on human-like meta-cognition. McFIS consists of two components: a cognitive component which is realized by a Takagi-Sugeno-Kang type neuro-fuzzy inference system and a meta-cognitive component which self-regulates the learning in cognitive component employing meta-cognitive knowledge. The meta-cognitive component ‘monitors’ the knowledge in the cognitive component

and ‘controls’ the learning. This McFIS has been employed to solve various regression and time-series prediction problems. The experimental studies clearly shows that McFIS outperforms other state-of-the-art neuro-fuzzy inference systems.

- **Meta-Cognitive Neuro-Fuzzy Inference System based Classifier:** The above developed McFIS is extended to solve classification problems, efficiently (McFIS-classifier). In the proposed classifier, class-specific criteria are employed during meta-cognitive monitoring of the knowledge. In addition, the learning algorithm also considers class-wise knowledge while initializing and updating the fuzzy rules. While allocating a new rule, the rule overlap as well as the output weights is initialized such that misclassification is minimized. Moreover, while initializing the output weights, the learning algorithm penalizes the inference system for letting the wrong class get close to correct output.
 - **Application to Video Analytics:** In this thesis, we employ the proposed McFIS-classifier for two video analytics problems of human action recognition and facial emotion recognition.
 - **Human Action Recognition:** An efficient action recognition system from video based on meta-cognitive neuro-fuzzy inference system is proposed in this thesis. In order to represent actions, optical flow based features, which can represent information from local pixel level to global object level, is employed. The aim of McFIS-classifier is to find the optimal decision boundary separating each action based on the optical flow based features extracted from different video sequences. The performance of the proposed action recognition system is evaluated on two benchmark data sets.
- In order to efficiently classify actions based on 3D video sequences, a 3D action recognition system using McFIS-classifier is proposed. Here, the depth flow information from 3D video is combined with optical flow based features to form 3D optical flow based features. The performance of the 3D action recognition system is evaluated on a proprietary human action recognition data set.

- **Facial Emotion Recognition:** In this thesis, we study the performance of various feature extraction techniques reported in literature for facial emotion extraction and employ them to evaluate the generalization ability of McFIS-classifier. In here, pixel based, Gabor filter based, scale invariant feature transform based, curvelet based and local binary pattern based features are extracted from two benchmark data sets. Each of these feature extraction techniques have their own advantages. We conduct a stratified 10-fold cross-validation study to evaluate their advantages. Further we evaluate the generalization ability of each of these features on recognizing human emotions by conducting a person independent emotion recognition study and data set independent emotion recognition study.

1.5 Thesis Organization

This thesis is organized as follows (refer Fig. 1.2):

- In chapter 2, existing neuro-fuzzy inference systems and their learning algorithms are reviewed based on the types of inference mechanism, types of learning, learning modes, and applications in which they are employed.
- Chapter 3 presents a Meta-Cognitive sequential learning algorithm for a Takagi-Sugeno-Kang type Neuro-Fuzzy Inference System (McFIS) for function approximation. The learning mechanism controls *what-to-learn*, *when-to-learn* and *how-to-learn* by efficiently deciding on *sample deletion strategy*, *sample reserve strategy*, and *sample learning strategy*, respectively. Moreover, the learning algorithm fully exploits the localization property of Gaussian rule during rule addition. The performance of the developed McFIS is evaluated on a set of four benchmark function approximation and time-series prediction problem. The performance of McFIS has been compared in terms of training and testing error, number of rules grown and number of samples employed for training. It is found that McFIS is able to approximate the underlying functional relationship between the input and the output efficiently, employing lesser number of rules and fewer samples than other state-of-the-art neuro-fuzzy inference systems.

Figure 1.2: Thesis Organization

- In chapter 4, the developed McFIS is extended to solve classification problems, efficiently. In the developed classifier, class-specific criteria are employed during meta-cognitive monitoring and also during rule evolution. The performance of McFIS based classifier is compared with a set of benchmark binary as well as multi-class data with balanced and imbalanced sample distribution. The performance is also evaluated on a set of real-world data sets. A quantitative performance analysis based on overall and geometric mean training/ testing accuracy, number of samples employed for training, and number of rules grown, was performed. A qualitative one-way repeated measures ANOVA followed by a *post-hoc* Bonferroni-Dunn test is also performed. It could be clearly observed from the performance analysis that McFIS performs better than other state-of-the-art classifiers reported in the study.
- McFIS based classifier is employed to solve video analytics problem of human action recognition, in chapter 5. Assuming the task of action recognition as a classification problem, the aim of McFIS based classifier is to identify human actions based on action representation. In this thesis, optical flow based features are employed for efficient action representation, since they can represent apparent movement in objects from local pixel level to global object level. The performance of McFIS based action recognition system is evaluated on a set of real-world action recognition data set.

McFIS based action recognition system is further extended to efficiently detect 3D actions from video sequences. In order to represent 3D actions, 2D optical flow based features are combined with depth flow features. The performance of 3D action recognition system is evaluated on a set of proprietary action recognition data set. The performance evaluation on 2D as well as 3D action recognition problems indicate superior performance of McFIS.

- In chapter 6, McFIS based emotion recognition system is developed. Ability of the proposed emotion recognition system is evaluated based on various feature extraction techniques reported in literature. Further, this thesis also evaluate the ability of McFIS to identify emotions, independent of the person as well as data set. The performance evaluation on two benchmark emotion recognition data set

indicates the improved performance of McFIS based emotion recognition subsystem over other state-of-the-art classifiers.

- Chapter 7 summarizes the conclusions from the study and presents the recommendations for future works.

Chapter 2

A Review of Existing Neuro-Fuzzy Inference Systems and Their Learning Algorithms

2.1 Introduction

In artificial intelligence community researchers aim to endow machines with human like traits such as ability to learn and reason. As a result, various computing models (such as artificial neural networks, fuzzy system) emulating different parts of brain and their functionalities have been proposed [76, 80, 136, 31]. These neural networks can accurately model the functional relationship between the input and the output, by virtue of the parallelism, connectionism and learning mechanism, emulated from different parts of the brain.

In the literature of artificial intelligence, research works are also carried out to empower machines with human-like ability to represent information more accurately and also to tolerate impreciseness [129]. Fuzzy logic theory [254] is one of the most significant developments in this direction. It enables a machine to mimic the human ability of tolerating imprecision and uncertainty, by the use of fuzzy rules and fuzzy inference mechanism. The two techniques of artificial neural networks and fuzzy logic based systems have been employed extensively in literature of system control and soft-computing [86, 205]. Moreover, works are also available in literature which combine these two technologies to achieve the twin requirements of accuracy and imprecise knowledge representation. In [73, 214] authors have roughly categorized these combinations at their

Figure 2.1: Fusion Level Categorization of Artificial Neural Networks and Fuzzy Inference Systems Adapted from [214]

fusion level perspective. A short description of various hybridizations of neural networks and fuzzy rule based systems, adapted from [214] is provided herein (refer Fig. 2.1):

- 2.1(a) Here, an artificial neural network and an fuzzy rule based system are independently employed in the same system. These fusion technologies are employed in systems where they are employed for different purposes and do not cooperate with each other [54].
- 2.1(b) In the first figure, the output of an artificial neural network and a fuzzy rule based system are combined. In Fig. 2.1(b), the output of a neural network or fuzzy system is modified with the output of a fuzzy system or neural network, respectively.
- 2.1(c) A fuzzy system and neural network are cascade connected, in these cases. Such fusion technologies have been employed in various signal processing applications [181, 216].
- 2.1(d) This hybridization helps to design or tune fuzzy rules using a learning algorithm derived from neural networks. One of the early applications of such a system was proposed in [74]. Similar learning techniques are now widely used in neuro-fuzzy inference systems.
- 2.1(e) This figure shows the network realization of a fuzzy inference system. Similar realization has been widely employed in many modern day neuro-fuzzy inference systems [169, 12].
- 2.1(f) These are the cases where a neural network configures fuzzy rules or conducts fuzzy reasoning. An example of such a system include cerebellar model articulation controller dealing with fuzzy rules [82].
- 2.1(g) Here, certain parts of a fuzzy rule based system are replaced by a feed-forward neural network or membership functions are adjusted by a neural network.
- 2.1(h) A fuzzified artificial neural network is depicted here. There include neural networks modified to handle fuzzy input, output and weights [91].

In this work a combination of hybridization employed in Figs. 2.1(d) and 2.1(e) is realized. An artificial neural network based algorithm is employed to tune the fuzzy rules of a neuro-fuzzy inference system.

In the literature of soft-computing it has been shown that a radial basis function neural network and a fuzzy inference system are functionally equivalent [94]. This equivalence allows one to employ the learning techniques and learning modes developed for artificial neural networks for tuning the rules of a fuzzy inference system. Such a fuzzy inference system employing learning algorithm inspired from or derived from artificial neural networks is called a neuro-fuzzy inference system. In literature, various neuro-fuzzy inference systems have been proposed [169, 171, 193] and they have been increasingly employed in various applications such as aircraft control [170], remote sensing [243], etc. In this chapter, the different neuro-fuzzy inference systems available in literature are discussed in detail, along with their limitations. As neuro-fuzzy inference systems are fusion of fuzzy inference system and neural network based learning algorithm, they can broadly be classified based on several related criteria, as shown below:

- Classification based on the type of Fuzzy Inference Mechanism
 - Takagi-Sugeno-Kang fuzzy inference mechanism
 - Mamdani fuzzy inference mechanism
- Classification based on the type of Membership Function
 - Non-Differentiable membership functions
 - Differentiable membership functions
- Classification based on the type of Learning Mode
 - Batch/ Offline learning
 - Sequential/ Online learning
- Classification based on the Applications
 - System Control

– Soft computing

The literature of neuro-fuzzy inference systems could also be divided based on the type of input/ output signals as belonging to real valued network or complex-valued network. However, few works are available till date which deal with complex-valued neuro-fuzzy inference system and hence, not reviewed.

2.2 Fuzzy Inference Mechanisms

In the literature of fuzzy inference systems, various inference mechanisms have been proposed [215, 131, 14, 227]. Among these inference mechanisms, two main inference mechanisms have been widely used due to the interpretability and accuracy required for an application: Mamdani fuzzy inference mechanism [131] and Takagi-Sugeno-Kang inference mechanism [215].

2.2.1 Mamdani Fuzzy Inference Mechanism

Mamdani fuzzy inference mechanism [131], was proposed by E. H. Mamdani in 1974. One of the advantages of such an inference mechanism is its rule interpretability, but lacks accurate solution formulation ability. According to this inference mechanism, for any fuzzy system with m input features, n output features and K rules, the k^{th} rule, R_k , is expressed as

$$R_k : \text{ IF } x_1 \text{ is } A_{1k} \text{ AND } x_2 \text{ is } A_{2k} \text{ AND } \cdots \text{ AND } x_m \text{ is } A_{mk} \quad (2.1)$$

$$\text{ THEN } y_j = B_{jk} \quad (2.2)$$

where, $\mathbf{x} = [x_1, \cdots, x_m] \in \mathbb{R}^m$ are the m -dimensional input features and $\mathbf{y} = [y_1, \cdots, y_n] \in \mathbb{R}^n$ are the n -dimensional output. $\mathbf{A}_k = [A_{1k}, \cdots, A_{mk}] \in \mathbb{R}^m$ are the fuzzy rule antecedents of k^{th} rule and $\mathbf{B}_k$ is the corresponding consequent. Each of the k rules are combined employing a T-conorm operator.

In a neuro-fuzzy inference system employing Mamdani fuzzy inference mechanism, the first step is to determine the membership of the current input with respect to the

rules in the network. The membership of i^{th} input feature of a sample $\mathbf{x}$ with respect to k^{th} rule is given by

$$\phi_{ik} = \exp\left(\frac{(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}\right), \quad i = 1, \dots, m \quad (2.3)$$

It must be noted that μ_k and σ_k are the center and width of the k^{th} Gaussian rule employed. Instead of Gaussian membership function, one can employ other radial basis function as well, during fuzzification. The next step is aggregation operation of all the individual memberships. A minimum T-norm operator is employed for feature aggregation, and it is given as,

$$\phi_K = \text{T-norm}(\phi_{1k}, \dots, \phi_{mk}) \quad (2.4)$$

$$= \min(\phi_{1k}, \dots, \phi_{mk}) \quad (2.5)$$

The fuzzy inference system combines the aggregated rule antecedents and determine the degree to which they belong to the output. It is achieved by using a S-norm operator. Here, a maximum S-norm is employed. The output of a Mamdani fuzzy inference system is computed by employing techniques such as weighted average defuzzification, center of gravity defuzzification, etc.

Various works are available in the literature of neuro-fuzzy inference systems, which employ Mamdani fuzzy inference mechanism [41, 165, 147, 182]. Next, we shall describe Takagi-Sugeno-Kang fuzzy inference mechanism.

2.2.2 Takagi-Sugeno-Kang Fuzzy Inference Mechanism

Takagi-Sugeno-Kang type fuzzy inference mechanism was first employed in [215]. Let the fuzzy system consist of m input features, n output features and have K rules. The k^{th} rule for such a system, R_k , is expressed as

$$R_k: \text{ IF } x_1 \text{ is } A_{1k} \text{ AND } x_2 \text{ is } A_{2k} \text{ AND } \dots \text{ AND } x_m \text{ is } A_{mk} \quad (2.6)$$

$$\text{ THEN } y_j = \alpha_{0jk} + \alpha_{1jk}x_1 + \dots + \alpha_{njk}x_n \quad (2.7)$$

where, $\mathbf{x} = [x_1, \dots, x_m] \in \mathbb{R}^m$ are the m -dimensional input features and $\mathbf{y} = [y_1, \dots, y_n] \in \mathbb{R}^n$ are the n -dimensional output. $\mathbf{A}_k = [A_{1k}, \dots, A_{mk}] \in \mathbb{R}^m$ are the fuzzy rule antecedents of k^{th} rule and $\boldsymbol{\alpha}_k = [\alpha_{0jk}, \dots, \alpha_{njk}]$ are the consequent parameters of the k^{th} rule. Each rule is combining using an S-norm operator.

Fuzzy Inference Systems employing Takagi-Sugeno-Kang inference mechanism represents the non-linearity in a given function as a piecewise weighted linear combination of inputs. It has the ability to represent a network accurately with lesser rules, but at the cost of decreased interpretability due either to the way rules are combined or to the high number of consequent parameters leading to high dimensionality. In any such neuro-fuzzy inference system, the first step is to determine the membership of the current input with respect to the rules in the network. The membership of i^{th} input feature of a sample $\mathbf{x}$ with respect to k^{th} rule is given by

$$\phi_{ik} = \exp\left(\frac{(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}\right), \quad i = 1, \dots, m \quad (2.8)$$

It must be noted that μ_k and σ_k are the center and width of the k^{th} Gaussian rule employed. Instead of Gaussian membership function, one can employ other radial basis function as well, during fuzzification. The next step is aggregation operation of all the individual memberships. In Takagi-Sugeno-Kang fuzzy inference mechanism, a product T-norm operator is employed for feature aggregation. It is given as

$$\phi_k = \text{T-norm}(\phi_{1k}, \dots, \phi_{mk}) \quad (2.9)$$

$$= \phi_{1k} \times \dots \times \phi_{mk} \quad (2.10)$$

Takagi-Sugeno-Kang type fuzzy inference mechanism can be split into two, based on how the consequent parameters are modeled: type-0 and type-1 inference mechanism. In type-1 inference mechanism, the consequent parameters are modeled as a linear combination of inputs and output weights. Consequent parameters for k^{th} rule is given by,

$$v_k = \alpha_{0k} + \alpha_{1k}x_1 + \dots + \alpha_{mk}x_m \quad (2.11)$$

In the use of type-0 inference mechanism, $\alpha_{1k}, \dots, \alpha_{mk}$ are set to zero. As a result

$$\alpha_{0k} = \phi_k \quad (2.12)$$

The output of a Takagi-Sugeno fuzzy system is a weighted center of gravity defuzzified output, given as

$$\hat{y} = \frac{\sum_{k=1}^K v_k \cdot \alpha_{0k}}{\sum_{k=1}^K \alpha_{0k}} \quad (2.13)$$

In literature of neuro-fuzzy inference system, various research are available which employ Takagi-Sugeno-Kang fuzzy inference mechanism [12, 126, 197].

2.2.3 Other Inference Mechanisms

In literature, various other inference mechanisms have also been proposed. In [224], an inference mechanism is proposed which aims at reducing the time complexity by avoiding defuzzification. In this inference mechanism, the antecedent rule aggregation is modeled employing T-norm operation, whereas, the consequent of each rule is represented by a fuzzy set with a monotonical membership function. As a result, the output of each rule is a crisp value induced by rule's firing strength. The overall output is calculated as the weighted average of each rule's output, thereby saving computational cost involved. However, this inference mechanism is not transparent as Mamdani or Takagi-Sugeno-Kang inference mechanisms.

Inference mechanisms have also been proposed with the aim of improving the accuracy as well as interpretability of a fuzzy inference system. One such inference mechanism has been proposed in [105]. Here, a heuristic fuzzy logic rule and input-output fuzzy membership function is employed for solving time-series identification and regression problems. The rule membership is modeled by a Gaussian membership function, and the rule antecedent employs an *AND* T-norm operator to combine each of the individual memberships. The rule consequents employ *OR* S-norm operator to calculate the degree to which the product of antecedent and output weights are supported by all fuzzy rules together. The output of the network is calculated employing a center of area defuzzification technique.

Another fuzzy inference mechanism aiming at attaining accuracy and interpretability is proposed in [227, 78]. These approaches combine Takagi-Sugeno-Kang and Mamdani inference mechanisms. Here, the rule antecedents are calculated similar to that of Mamdani fuzzy inference mechanism. Two rule consequents are calculated corresponding to Mamdani and Takagi-Sugeno-Kang fuzzy inference mechanism. The total output of the system is calculated by combining the inferred output of both the fuzzy rule consequents.

In [14], an inference mechanism based on the concept of data cloud has been proposed. This system is realized as a fuzzy aggregation of multiple loosely connected local systems. Such a formulation helps the system in avoiding the issues related to the choice of membership functions and its parametric representation. The data cloud is recursively updated enabling the system to work in both offline as well as online learning mode.

In the next section, we shall classify fuzzy inference systems based on the membership function they employ.

2.3 Membership Functions

In the previous section, a review of existing fuzzy inference mechanism was provided. In this section, we review major membership functions employed in the literature of fuzzy inference systems. In any fuzzy inference system, one of the most important components are the rules. They represent the knowledge extracted by the system.

The extraction of rules requires partitioning of the input domain. A few techniques to partitioning of input space is grid partitioning, clustering and hierarchical splitting [190]. In the grid partitioning and clustering, the input attributes are divided into bounded intervals, and hence it is natural to apply any radial basis functions, such as Gaussian function. In hierarchical domain partitioning, the use of half-bounded sigmoidal functions are reported to be employed [190]. In literature, various activation functions have been proposed with the aim of providing a unique membership for a given input. These membership functions could be broadly classified as either non-differentiable or differentiable. We employ such a classification, to help the selection of learning algorithms for the neuro-fuzzy inference systems.

2.3.1 Non-Differentiable Membership Functions

One of the first fuzzy rules employed trapezoidal basis function, and triangular membership functions. A trapezoidal membership function is defined by the four vertices of a trapezoid, a , b , c , and d . In a triangular membership function, $b = c$. The membership of an input x for a rule given by variables a , b , c , and d employing trapezoidal membership function is given by

$$\text{trapezoid}(x; a, b, c, d) = \max \left(\min \left(\frac{x - a}{b - a}, 1, \frac{d - x}{d - c} \right), 0 \right). \quad (2.14)$$

Similarly, for a triangular basis function with input x and rules a , b , d , it is given by

$$\text{triangle}(x; a, b, d) = \max \left(\left(\frac{x - a}{b - a}, \frac{d - x}{d - b} \right), 0 \right) \quad (2.15)$$

It could be observed that such polygonal membership functions require small amount of data to define a membership function and it is easy to obtain a mapping of input-output space as a hyper-surface of linear segments.

In literature, various fuzzy inference systems have successfully employed these membership functions [43, 130]. In [130], an isosceles triangle fuzzy set is used, whereas in [43], a moving isosceles triangle fuzzy set is employed. In [113], a trapezoidal membership function is employed in designing an evolving neural network. Neuro-fuzzy inference systems employing these non-differentiable membership functions use non-error based optimization techniques such as evolutionary algorithms, simulated annealing or local optimization techniques to tune the fuzzy rules. However, one of the main disadvantages of these membership functions is that they are not continuous. This renders them non-differentiable and as a result, smooth adaptation of rules is not possible.

2.3.2 Differentiable Membership Functions

In the literature of neuro-fuzzy inference systems, the use of radial basis functions as membership functions are very common, especially due to their easy of design, generalization ability, tolerance to input noise and online learning ability [252]. Moreover, they are continuous, which lends them applicable in various gradient based learning algorithms derived from neural networks. A commonly employed radial basis membership function is Gaussian function, which is defined in terms of its center and width. The membership of an input x , employing a Gaussian membership function with center μ and spread σ is given as

$$Gaussian(x; \mu, \sigma) = \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (2.16)$$

This function is fully differentiable and satisfies monotonicity, itself. Similar membership functions have been employed in various recent neuro-fuzzy inference system [197, 256, 259].

In literature, authors have also employed B-spline function as membership functions in a fuzzy inference system. It has been shown that B-spline network can approximate any continuous functions at any arbitrary accuracy. One such neuro-fuzzy inference system employing a B-spline membership function is reported in [87]. In here, an algebraic method for incremental learning is adapted from neural networks is employed for learning

a fuzzy decision tree with B-spline membership function, represented as a three layered neural network. These inference systems have been shown to perform well on regression and time series prediction problems. However, these membership functions cannot represent many simple curves such as circle or ellipses.

Neuro-fuzzy inference systems employing wavelet based membership functions have also been reported in literature. The major advantage of employing wavelet membership function [28] is the existence of local support and provision of compact local representation of signals in both frequency and time domain, in addition to the presence of scaling and shifting parameters to process signals with local features. One of the recent works employing wavelet membership function has been reported in [28]. Here, a Mexican Hat wavelet is employed as basis function. The membership function is defined by three parameters, center c , width σ and function form parameter α . The membership Φ for an input x is given by

$$\Phi(\tau_k) = (1 - \alpha\tau_k^2) \exp\left(-\frac{\tau_k^2}{2}\right) \quad (2.17)$$

where

$$\tau_k = \left((x - c(k))^T (\sigma^T \sigma)^{-1} (x - c(k))\right) \quad (2.18)$$

Authors in [28] employ a Takagi-Sugeno-Kang fuzzy inference mechanism and a gradient descent based learning algorithm for tuning the fuzzy rules. Moreover, in the above given membership function the function form parameter α controls the shape of the membership function. The choice of $\alpha = 0$ will provide a conventional Gaussian membership function, whereas $\alpha = 1$ will result in a Mexican hat wavelet membership function.

In the literature of artificial neural networks, research works are available where a q -Gaussian function [55] is employed as an activation function. Such an activation function parameterizes standard Gaussian distribution by replacing exponential expressions with q -exponential expressions. The q parameter allows representation of different basis function, such as Cauchy, inverse multi-quadratic, Gaussian, etc. and thus help the function match the distribution of distance more accurately. The q -Gaussian membership ϕ , with parameters center (μ) , width $(\sigma(k))$ of an input x is given by,

$$\phi(x; \mu, \sigma, q) = (1 + (1 - q)d(x, \mu, \sigma))^{\frac{1}{(1-q)}} \quad (2.19)$$

where,

$$d(x, \mu, \sigma) = (x - \mu)^T (\sigma^T \sigma) (x - \mu) \quad (2.20)$$

Such an activation function can also be employed in a neuro-fuzzy inference system.

Recently in [14], a cloud based fuzzy set representation has been proposed for an evolving neuro-fuzzy inference system. This representation of fuzzy set helps remove the assumption with regard to particular cluster shape or its parameters. The degree of membership to the cloud is found as a function of its density. The membership calculation employs different basis function such as Cauchy basis function.

In this work we employ a Takagi-Sugeno-Kang fuzzy inference mechanism with a Gaussian membership function. A review on neuro-fuzzy rule generation is available in [137]. We shall critically review the major learning techniques and modes employed in existing neuro-fuzzy inference system, in the following section.

2.4 Learning Modes

In the earlier sections, the literature in neuro-fuzzy inference systems were reviewed based on the fuzzy inference mechanism employed and also based on the choice of fuzzy membership function. In this section, we shall review the literature in neuro-fuzzy inference systems on whether the training samples are available *a priori* or not. Accordingly, there are mainly two learning modes when applying learning algorithms derived from neural networks to fuzzy inference system, batch learning mode and sequential learning mode.

In both batch learning as well as sequential learning modes, we assume that a supervised learning technique is applied. Although there are research works reported in the literature of unsupervised learning, such as clustering, dimensionality reduction, feature selection, which fuzzify inputs and employ fuzzy memberships functions as a distance measure, these tasks does not rely on fuzzy inference mechanism, and as a result on fuzzy inference system.

2.4.1 Batch/ Offline Learning Mode

In offline learning algorithms, all the training samples are assumed to be available *a priori*. In batch learning mode, samples are assumed to arrive in batches and the network

parameters are adapted based on it. For the sake of classification convenience, in this chapter, all batch learning algorithms are assumed to be offline learning algorithms, with the structure of the network being fixed and the rules being initialized randomly or based on clustering approaches. The batch learning algorithms with evolving structure or adaptive parameters alone are classified as pseudo-online learning algorithms. Some of the initial works in these employing such a learning mode in neuro-fuzzy inference systems are neural-network-based fuzzy logic control system (FALCON) [118], adaptive network based fuzzy inference system (ANFIS) [93], neuro-fuzzy control (NEFCON) [142].

FALCON [118] is a five-layered network with an input layer, a fuzzification layer, a rule pre-condition layer, rule consequent layer and an output layer. The output nodes contain two linguistic nodes, one for each desired and predicted output. This network employs a combination of unsupervised and supervised learning approach. The initial rule base is initialized employing an unsupervised learning approach, and a gradient descent based learning is employed to adjust the parameters of the membership functions.

In [93], an adaptive network based fuzzy inference system (ANFIS) was proposed which implements a Takagi-Sugeno-Kang fuzzy inference mechanism in a five layered network architecture. It uses back propagation learning to determine the antecedent parameters and least mean square estimation to determine the consequent parameters. Learning in ANFIS includes two parts. In the first part, the antecedent parameters are assumed to be fixed, and the consequent parameters are estimated by a recursive least squared method. In the second part, the antecedent parameters are adjusted while maintaining the consequent parameters as constant. The procedure is iterated till convergence.

In literature, various neuro-fuzzy inference systems have also been proposed which employs variants of gradient descent based learning and least squares method. In [244], a Takagi-Sugeno-Kang neuro-fuzzy inference system has been proposed which employs a modified gradient descent based algorithm in reduces the cost of gradient calculation in comparison to standard gradient descent. In [235], a Takagi-Sugeno-Kang neuro-fuzzy inference system with ellipsoidal basis function has been proposed. A linear least squares method is employed to update the consequent parameters. Whereas, a neuro-fuzzy classifier for two class problem has been proposed in [62]. Here, fuzzy clustering based on

Gaussian mixture models and analysis of covariance is performed for initialization of fuzzy rule base. An orthogonal least square algorithm is employed for tuning the parameters of the network.

In NEFCON [142], a Mamdani fuzzy inference system is implemented where the learning process employs a combination of reinforcement learning and back propagation. Two variants of this system has been proposed for function approximation (NEFPROX) and classification (NEFCLASS). In [206], a fuzzy net was proposed, where a stochastic learning technique is employed to randomly change the parameters of membership function and connections within the network structure. The network is evaluated upon a random modification employing a cost function. The modification is reverted back if the performance degrades.

In recent times another group of learning algorithms for a single-layered feed forward neural networks, known as extreme learning machines, have been proposed in literature. Here, the hidden parameters of the network are randomly assigned and the output weights are analytically determined. These extreme learning machines have been successfully extended to fuzzy domain, e.g., [258, 207]. In [258], an extreme learning machine with fuzzy membership is proposed. In [207], fuzzy extreme learning machine employs K-means clustering to initialize the rule centers. However, it employs all the samples to learn the rule and hence, it is a purely batch learning algorithm.

Research works are also available in literature, which employ global or local optimization ability of evolutionary algorithms for funding the rules of the network. An evolutionary design of neuro-fuzzy systems has been proposed in [5] where a five-tier hierarchical evolutionary algorithm is employed to choose the membership function, architecture and learning parameters. The evolutionary neuro-fuzzy model can adapt either Mamdani or Takagi-Sugeno-Kang inference mechanism. The evolutionary algorithm chooses optimal type and value of membership functions between layers. Defuzzification and aggregation operators are adapted according to the fuzzy inference system by the evolutionary algorithm. In [119], a self-evolving evolutionary learning algorithm for neuro-fuzzy inference system is proposed for a Takagi-Sugeno-Kang fuzzy inference mechanism. A symbiotic evolutionary scheme is employed for structure identification and variant of particle swarm optimization of parameter tuning. In [35], an evolutionary computation

approach for finding the optimal learning rate of an evolving fuzzy neural network in a self-adaptive manner. Since these algorithms generate large population for finding optimal parameters, they are employed in an offline fashion. A self-organizing scheme for learning a Takagi-Sugeno-Kang neuro-fuzzy inference system was proposed in [115]. A geometric growing criterion is employed for selecting the initial rule parameters. A hybrid algorithm based on genetic algorithms, back propagation and recursive least squares estimation is employed to tuning the rules based on all the samples. A genetic algorithm based approach is employed for rule pruning.

In various practical problems, the training data arrive sequentially and the network has to adjust the parameters according to the *drifts* and *shifts* in the data [126]. Moreover, for problems in medical analytics, the number of samples for learning might be less. For such problems, batch/ offline training mode is not suitable.

In literature of neuro-fuzzy inference systems, algorithms have also been proposed wherein, the number of rules need to be specified *a priori*. In these algorithms, the rule antecedents and consequents are evolved from the training data. In this thesis, such algorithms are also brought under the ambit of sequential/ online algorithms. In sequential learning algorithms, the training samples arrive one-at-a-time and are discarded after learning. In the next section, we shall review sequential and online learning algorithms.

2.4.2 Sequential/ Online Learning Mode

In the previous section, a review of the major batch/ offline learning algorithms was presented. In this section, we shall present a review on sequential and online learning algorithms. Under this head, we also review evolving and adaptive algorithms for neuro-fuzzy inference systems. In an online training algorithm, all the samples to be learnt are not assumed to be available *a priori*. The samples arrive one-at-a-time. A Sequential learning mode stipulates that the training data is presented to the system one-by-one and be discarded after learning. Evolving learning algorithms are online version of adaptive learning mechanism, wherein the network is initially assumed to contain no rules, and the rules are grown and adapted based on the training data.

Initially, various learning algorithms were proposed for neuro-fuzzy inference systems, wherein, the structure learning or parameter learning were carried out sequentially or

adaptively, but independent of each other. In a few of these learning algorithms, the total number of samples used for learning also need to be specified. In this thesis, these algorithms are classified as pseudo-online learning algorithms.

One of the first algorithms employing pseudo-online learning modes were under the ambit of self-organized learning [97, 114, 120]. In [97] a self-constructing neuro-fuzzy inference system employing a Takagi-Sugeno-Kang fuzzy inference mechanism was proposed. Here, the structure and parameter identification occur simultaneously. An aligned clustering method is used for input space partitioning, where the rule antecedents are added based on nearest neighbor heuristic. Each of the generated rule is then decomposed into corresponding 1D membership function function, for both input and output space partitioning. The consequent parameters are initialized with a singleton value selected by a clustering method. A projection based correlation measure is used to select additional significant terms to be added to the consequent part. The consequent parameters are tuned using a least mean squares or recursive least squares technique, whereas the antecedents parameters are tuned using back propagation based learning. However, this method lacks method to prune excess rules. As a result, redundant or disfunctional rules are maintained in the network, leading to sub-optimal generalized performance with a large structure.

In [114], a self-organizing fuzzy neural network was proposed where the rules are extracted in an online fashion. The parameter tuning is carried out employing a recursive least squares method. This algorithm addresses the issue of rule pruning based on optimal brain surgeon technique, wherein a second derivative is employed to find unimportant rules. A self-organizing fuzzy neural network employing an extended Kalman filtering scheme is proposed in [120], which employs an online learning scheme with extended Kalman filter for updating parameters of the network. However, both the above mentioned networks require all the training samples for pruning. Hence, they cannot be classified as purely sequential.

In the domain of extreme learning machines, an online sequential fuzzy extreme learning machine was proposed in [168], based on Takagi-Sugeno-Kang fuzzy inference mechanism. Here, the number of rules have to be specified initially. The learning process consists of two phases: an initialization phase and a learning phase. During initialization phase, the network is trained with as many samples as the number of rules specified.

During the learning phase, the rules are tuned using a recursive least squares mechanism. However, the random initialization of the rule centers destroy the interpretability of a fuzzy inference system.

In the literature of neuro-fuzzy inference systems, few works initially addressed the issue of developing evolving architectures for neuro-fuzzy inference system. One of the first works to adaptively tune the rules of a neuro-fuzzy inference system is evolving fuzzy neural networks [100, 101]. This concept was further modified for classification problems in [65, 16, 11] and similar works. The system employed in [100, 101] uses an evolving connectionist system framework for adaptively tuning the rules of the network. The antecedents of the rule are adapted via a supervised/ unsupervised learning, whereas the consequent parameters are tuned using a supervised learning scheme. In [233], a mapping-constraint agglomerative clustering algorithm is employed for input space clustering. A recursive least squares method is employed for tuning the consequent parameters. However, these systems require the number of rules to be specified initially, as a result, not purely evolving.

Dynamic fuzzy neural networks [241] employs a Takagi-Sugeno-Kang based extended radial basis function neural network. Here, a hierarchical online self-organized learning scheme is proposed. The antecedent parameters of the network is evolved from the data, however, the parameter tuning employs a linear least squared estimation technique, and hence the learnt samples cannot be discarded. As a result, the network is not purely sequential in nature. In [242] authors provide some explanation on selection of learning parameters for the algorithm proposed in [241]. However, selection of these parameters require a knowledge of the total number of training samples. Hence, the learning cannot be purely online.

A dynamic evolving neuro-fuzzy inference system has been proposed in [193]. It employs an evolving clustering based algorithm for rule addition, which is based on Euclidian distance between the current sample and existing rules. The parameters are updated based on recursive least squares based method. In here, the rule are evolved based on all the training samples or selected samples. However, during the parameter tuning, the structure is kept fixed and the parameters of the network are tuned separately. Hence, this algorithm cannot be classified as purely online or evolving.

In a purely evolving and adaptive learning, the network is initialized with no rules, and the rules are grown and adapted based on the data, such that the final network is an efficient generalization of functional relationship between input and output. Here, the algorithms tune the rule antecedents and consequents simultaneously.

Angelov et. al. in [12] have proposed an evolving Takagi-Sugeno fuzzy model, which employs a first order Takagi-Sugeno-Kang fuzzy inference mechanism. It employs a variant of subtractive clustering called mountain clustering for rule clustering. Accordingly, potential is defined as a measure of spatial proximity between a given sample and existing rules. In this work, if the potential calculated is greater than a threshold, sample is said to be novel and a new rule is added to capture this novel knowledge. An extended Kalman filter is employed for tuning the rules of the network. However, this method lacks a rule pruning strategy. This issue has been addressed in [13], where a simplified evolving Takagi-Sugeno fuzzy model has been proposed. Here a scatter measure of the density of samples is used to cluster the input, as it is computationally cheaper in comparison to potential calculation. A cluster is neglected if the density of population is less than 1% of the total data samples. However, the cluster is only ignored in the current iteration. The rules are not pruned from the network, leading to increased computational effort. This issue has been rectified in [16] with the proposal of a rule pruning mechanism. Various other variant of this algorithm could also be found in literature [15, 17, 16, 125, 173, 126, 192]. In addition, the quality of the generated clusters are evaluated by calculating the number of samples under the influence of a given rule, its age in terms of the data samples. The parameters are updated based on weighted recursive least squares method.

Variants of the above evolving algorithms for classification has also been proposed in literature. In [16, 17, 125], authors present three evolving fuzzy rule based classifiers. In [16] two variants of an evolving classifier from data stream was proposed which employs a Takagi-Sugeno-Kang fuzzy inference mechanism. The main difference between the two is the order of the inference mechanism. The first work ‘eClass0’ employs a Takagi-Sugeno-Kang type-0 inference mechanism, whereas the second work ‘eClass1’ employs a type-1 variant. In order to solve classification problems efficiently, authors employ ‘per class’ measures during learning. A recursive density based fuzzy partitioning mechanism

is employed in eClass0 for rule generation, which employs potential based on cosine distance for determining the partitions. In eClass1, a global potential is employed for rule determination, as opposed to per class based measure. Once the antecedents are formed, consequents are tuned employing a recursive least squares based formulation. In [125], an evolving fuzzy classifier employing a winner-take-all inference mechanism is proposed. Here, an evolving vector quantization based technique is employed for fuzzy clustering and a recursive weighted least squares estimation method is employed for consequent parameters determination.

In [126] an evolving algorithm to detect and react to *drifts* and *shifts* in data samples is proposed by incorporating ‘age’ of a rule and including a ‘forgetting’ factor. For detecting *drifts/shifts* the concept of ‘age’ [13] has been employed. When a sample arrives, the age of a rule is changed. The magnitude of change depends on the how much the sample belongs to the rule. If a sample does not belong to a rule, its age is increased by a factor 1. Otherwise, it is increased by a smaller factor. If *drift/shift* is detected, a forgetting factor is introduced which changes the antecedents and consequent parameters of rule. In [192], a fast evolving neuro-fuzzy model has been proposed. In this model, a recursive Gatha-Geva clustering is employed to take advantage of the elliptical clusters used as the membership function. An inverse singular value decomposition is employed for reducing the computational cost involved in elliptical clusters.

A purely sequential learning algorithm, wherein samples are presented to the network only once and then discarded, called sequential adaptive fuzzy inference system was proposed in [169]. It employs a Takagi-Sugeno-Kang fuzzy inference mechanism. The rules are added and pruned from the network based on the influence of the rule calculated for each sample. An extended Kalman filtering scheme is employed for updating the parameters of the nearest rule. In [168], authors extended the algorithm proposed in [169] to a classifier. Here, the influence of a rule has been modified to include information about the number of feature, and the computationally expensive extended Kalman filter is replaced by a recursive least squares based method.

In [72], a growing and pruning fuzzy neural network wherein the sensitivity of a rule with respect to the current sample is used as a measure of influence of a rule. Here, a computationally intensive Fourier decomposition of the variance of the output over past

samples is used for the growing/pruning criteria and a slow gradient descent method is used for updating the structure.

In self-organizing framework, in [173], a self-organizing fuzzy modified least squared algorithm is proposed. This algorithm employs a density based rule pruning (the number of times the rule is used in a window of samples) to prune the excess rules and is an online learning algorithm. In addition, this algorithm employs a uni-dimensional membership function with only two parameters thereby avoiding the singularity produced by the width criterion in antecedent part.

2.5 Applications

In literature, neuro-fuzzy inference systems have been employed for various applications. These applications can be broadly divided as control applications and applications in soft computing. The domain of system control includes, control of systems like air conditioners and fans, to control of more complex systems like traffic control and aircraft control. In soft computing, neuro-fuzzy inference systems find application in problems involving, regression, prediction, modeling and classification. Next we shall review neuro-fuzzy inference systems used in each of these areas.

2.5.1 System Control

Following the success of fuzzy controllers, neuro-fuzzy inference systems have been widely used in the domain of system control. They have found wide use in areas including control of electro-mechanical systems, automobiles, medicine, network traffic, etc. These applications are mainly due to the use of model-free control technique, ability to store knowledge and adaptively change the knowledge.

In [229, 228], authors present a DC motor controller employing a neuro-fuzzy inference system. Here, an fuzzy controller based on adaptive neuro-fuzzy inference system [93] is employed for minimizing overshoot and settling time following sudden changes in load torque. In [232], an adaptive fuzzy neural network controller is employed for high-precision position tracking, whereas in [231] a sliding-mode controller properties are realized in an intelligent control system for precise position tracking of a n -link robot.

In [146] a neuro-fuzzy controller was proposed for autonomous mobile robot navigation and multi-robot convoying problem. In [172], a controller has been proposed for robotic arms control.

A vision based neuro-fuzzy controller for an unmanned aerial vehicle was proposed in [160]. The controller employs two identical networks for pan and tilt motion control. In [37], an fuzzy neural network is employed to learn the inverse model of eight-rotor micro ariel vehicle in an online fashion. In the control of land vehicles, neuro-fuzzy inference systems have found use in transmission control of automobile. In [75], the manual gear-shift system is automated employing a neuro-fuzzy inference system. A wheel slip controller using a neuro-fuzzy inference system was proposed in [38], wherein the network learns in an online fashion.

In medical domain, neuro-fuzzy controller has found a few applications. In [223], a anesthetic gas controller was proposed which is based on Electroencephalography signal and heartbeat of the patient. An insulin controller for Type-1 diabetic patients was proposed in [67].

Neuro-fuzzy systems have found a wide variety of application in traffic control, in recent times. In [255], a neuro-fuzzy system for dynamic classification of vehicles to avoid collision was proposed. An approach to autonomously adapt the speed of a vehicle from human driver was proposed in [154]. Whereas, in [1], best route selection algorithm based on neuro-fuzzy inference system was proposed for avoiding traffic congestion. In the domain of data traffic handling, a neuro-fuzzy inference system for for rate control for available bit rate service class in asynchronous transfer mode networks is proposed in [191, 144].

2.5.2 Soft Computing

The use of neuro-fuzzy inference system in the field of soft computing is increasing day-to-day. In recent times, it have found numerous use in applications in various domains such as finance, medicine, security and surveillance, energy sector, etc.

2.5.2.1 Financial Prediction

In the field of finance, [161] has employed a self-organizing fuzzy neural network for portfolio management, whereas in [8] authors have proposed an adaptive neuro-fuzzy

inference system for portfolio analysis. Neuro-fuzzy inference systems have also found applications in predicting financial volatility [226], for options trading [152], for stock prediction [162], etc. Authors have also employed neuro-fuzzy based system for energy price forecasting [167].

2.5.2.2 Medical Analytics

Electroencephalogram (EEG) has become a basic tool for multitude of brain studies. However, classification of brain signals based on EEG is an open problem. In [18] a neuro-fuzzy inference system based approach was employed for selection of EEG channel in brain-computer interfacing, whereas in [247], an EEG classification method based on hidden markov model and an evolving classifier was proposed. It considers the spatial proximity between input signals while classifying. In [45], a similar approach was employed for fusion of medical images. In [23], a neuro-fuzzy based approach has been employed for Alzheimer's disease prediction. In medicine, adaptive neuro-fuzzy inference system [93] has been employed for numerous applications. A review of the same is available in [81]. A cervical cancer detection system based on infrared spectroscopy and an evolving classifier is proposed in [104, 103].

2.5.2.3 Video Analytics

In the area of security and surveillance, in [196] a human action recognition system is proposed employing a meta-cognitive neuro-fuzzy inference system, whereas an evolving fuzzy classifier for real-time human action recognition is proposed in [9]. A similar evolving classifier was employed for detection of human activities from wearable sensors was reported in [9, 88]. In [197], a meta-cognitive neuro-fuzzy inference system based classifier has been proposed for emotion recognition. In [138, 19], neuro-fuzzy inference system based approach has been employed for object tracking. In [60], a traffic monitoring system based on neuro-fuzzy inference systems have been proposed.

2.5.2.4 Energy Sector

Authors in [200], a complex-valued neuro-fuzzy inference system has been proposed for efficient wind speed and direction prediction, whereas in [156] ANFIS [93] has been

employed for wind turbine power coefficient estimation. Such systems help in efficient load scheduling. In [117, 151], neuro-fuzzy systems based technique has been used for efficient load forecasting.

2.6 Summary

In this chapter, various techniques of fusing fuzzy inference systems and artificial neural networks were initially presented. One of the fusion technologies employed neural network based learning algorithms to tune neuro-fuzzy inference systems is of special interest in this thesis. The literature of neuro-fuzzy inference systems and their learning algorithms were classified based on the type of inference mechanism, membership function, learning methods employed. A short summary of the major variants of the neuro-fuzzy inference systems reviewed is provided in tabular form in Table 2.1. The first works in the literature of neuro-fuzzy inference systems were more inclined towards the fuzzy aspect of learning. They employed non-differentiable membership functions and aimed at achieving more interpretability. In course of time, more works started to be conducted using differentiable membership functions, significantly due to the applicability of error correction learning, to improve accuracy. The increase in amount of data available for learning is leading to the use of online sequential learning algorithms where data are discarded upon learning once. Moreover, various works are currently being conducted towards combining interpretability of fuzzy systems and predictability of neural networks in a adaptive learning environment.

One of the main drawbacks of the current learning technology is the ill-effect of over-training. Meta-cognitive learning strategy which provides a network with the ability to self-regulate its learning needs to be incorporated in the learning algorithm. The aim of this thesis is to propose such a meta-cognitive learning algorithm for a neuro-fuzzy inference system.

Table 2.1: Summary of Major Types of Algorithms Discussed in Literature Review

Publication	Inference Mechanism	Membership Function	Learning Mode
Lin and Lee [118]	Hybrid (Functional inference)	Gaussian	offline (unsupervised + gradient descent)
Jang [93]	Takagi-Sugeno-Kang	Gaussian	Offline (gradient descent + recursive least squares)
Nauck et al. [143]	Mamdani	Gaussian	Offline (reinforcement + back propagation)
Ajith and Baikanth [5]	Hybrid (Mamdani or Takagi-Sugeno)	evolved	Offline (evolutionary)
Lin and Chen [119]	Takagi-Sugeno-Kang	Gaussian	Offline (sybiotic evolutionary algorithm + particle swarm optimization)
Wang et al. [234]	Takagi-Sugeno-Kang	Ellipsoidal	Offline (extended Kalman filter + least squares method)
Nowak and Nowicki [147]	Mamdani	Gaussian	Offline (gradient descent)
Scherer [182]	Mamdani	Gaussian	Offline (back-propagation)
Cpalka et al. [41]	Mamdani	Triangular	Offline (evolutionary)
Bodianskiy and Vynokurova [28]	Takagi-Sugeno-Kang	Wavelet	Offline (gradient descent)
Zhang and Ji [258]	Takagi-Sugeno-Kang	Gaussian	Offline (least squares method)
Leite et al. [113]	Hybrid (Functional + Linguistic)	Triangular	Offline (recursive least squares)
Wu and Meng Joo [241]	Takagi-Sugeno-Kang	Gaussian	Pseudo-online + adaptive (online clustering + least squares + offline parameter selection)
Juang and Lin [97]	Takagi-Sugeno-Kang	Gaussian	Pseudo-online (recursive least squares + back propagation)
Angelov and Filev [12]	Takagi-Sugeno-Kang	Gaussian	Pseudo-online (evolving clustering + recursive least squares)
Rong et al. [168]	Takagi-Sugeno-Kang	Gaussian	Pseudo-online (extreme learning machine + offline clustering)
Ho et al. [78]	Hybrid (Mamdani + Takagi-Sugeno)	Gaussian	Pseudo-online (online clustering + gradient descent)
Lin and Meng Joo [120]	Takagi-Sugeno-Kang	Gaussian	Pseudo-online + adaptive (Statistical + extended Kalman filter + of-fine pruning)
Rong et al. [169]	Takagi-Sugeno-Kang	Gaussian	Sequential (statistical + extended Kalman filter)
Angelov and Zhou [16]	Takagi-Sugeno-Kang	Gaussian	Online (evolving clustering)
Lughofer [125]	Takagi-Sugeno-Kang	Gaussian	Online (vector quantization + recursive weighted least squares)
Rubio [173]	Takagi-Sugeno-Kang	Gaussian	Online + sequential (modified least squares method)
Lughofer and Angelov [126]	Takagi-Sugeno-Kang	Gaussian	Online (recursive least squares for antecedent and consequent)

Chapter 3

A Meta-Cognitive Sequential Learning Algorithm for Neuro-Fuzzy Inference System (McFIS)

3.1 Introduction

In the previous chapter a review on major neuro-fuzzy inference systems and their learning algorithms was presented. This chapter focuses on the development of a meta-cognitive sequential learning algorithm for a Takagi-Sugeno-Kang type neuro-fuzzy inference system.

In the literature of neuro-fuzzy inference systems, various sequential learning algorithms have been reported [12, 169, 168]. These algorithms assume a uniform distribution of training data and captures the knowledge in these samples in the network. These algorithms address the aspect of *how-to-learn* the functional relationship between input-output, efficiently, from the training samples, in a sequential framework where samples are presented one-by-one and only once. In addition, existing algorithms determine the number of rules required to approximate the input-output relationship. However, these algorithms do not consider the problem of over-training upon repeatedly learning similar samples. In addition, it has been shown in [211] that altering the sequence of learning and skipping samples which contain similar knowledge affect the generalization ability of a network, positively. Thus, there is a need to develop a learning algorithm for neuro-fuzzy inference system which is able to self-regulate its learning by deciding *what-to-learn*, *when-to-learn* and *how-to-learn*, from training samples.

Figure 3.1: (a) Nelson and Narens Model of Meta-Cognition Adapted from [145] and (b) its Analogy in McFIS

Recent studies in educational psychology indicate that meta-cognitive learning that empowers the human learners with self-regulation is the best learning strategy [90]. Meta-cognition enables human learner to assess their current knowledge, identify when new knowledge is required and in addition, provide them with strategies to acquire the knowledge. In this chapter, we propose such a meta-cognitive sequential learning algorithm for a neuro-fuzzy inference system, referred to as Meta-Cognitive Neuro-Fuzzy Inference System (McFIS). The work done in this chapter has been published in [197].

In literature various models of meta-cognitions have been proposed, a review of which is available in [40] and has been summarized in chapter 1. Nelson and Narens [145] proposed a simple model of meta-cognition which is in line with studies in brain analysis and is highly compatible with the views presented in computational literature [40]. In this thesis, we develop McFIS based on well-known model of meta-cognition proposed by Nelson and Narens. According to Nelson and Narens' model of meta-cognition [145], cognitive processes could be split into at least two specifically interrelated components, referred to as cognitive component and meta-cognitive component as shown in Fig. 3.1(a). There are two main relationships 'control' and 'monitoring' which are defined in terms of direction of flow of signals between the cognitive component and the meta-cognitive component. The signal flowing from the meta-cognitive component to the cognitive component is 'control' signal and the basic notion underlying control is that the meta-cognitive component modifies the cognitive component based on these signals. In particular, the information

flowing from the meta-cognitive component changes either the state of the cognitive component or the cognitive component itself. Based on the control signal one of the three actions could occur at cognitive component: (1) item selection, (2) selection of kind of processing or (3) termination of study. Since the control signal in itself does not yield any information about the state of the cognitive component, a logically independent ‘monitoring’ signal is employed from cognitive component to meta-cognitive component. The objective of the ‘monitory’ signal is to inform the meta-cognitive component about the cognitive component. Based on the monitory signal, meta-cognitive component changes its knowledge about the cognitive component.

In line with Nelson and Narens model of meta-cognition, McFIS consists of two inter-related components: a neuro-fuzzy inference system is the cognitive component, whereas a self-regulatory sequential learning mechanism is the meta-cognitive component (refer Fig. 3.1(b)). The meta-cognitive component has a dynamic model of the neuro-fuzzy inference system. There are two type of signals: ‘control’ signal which originates from the self-regulatory learning mechanism and ‘monitoring’ signal which flows from neuro-fuzzy inference system to the meta-cognitive component. During the learning process, the self-regulatory learning mechanism ‘monitors’ the knowledge in the neuro-fuzzy inference system and ‘controls’ the learning in the same by changing the state of the cognitive component or the cognitive component itself. In this chapter, we employ instantaneous sample error and spherical potential [79] as the monitoring signals. The self-regulatory learning mechanism selects the suitable learning component, from *what-to-learn*, *when-to-learn* and *how-to-learn*, for each sample. These three components are realized in McFIS by the three strategies: *sample deletion*, *sample learning* and *sample reserve* strategies.

To summarize, McFIS activates one of the following three strategies during learning:

- (a) *Sample deletion strategy*: Samples with similar information content are deleted without being learnt.
- (b) *Sample learning strategy*: Depending on the information content in the current sample with respect to existing knowledge, the meta-cognitive component initiates rule growing, parameter update or rule pruning actions. During rule growing, a rule is added to McFIS such that the newly added Gaussian rule exploits the localization

property completely. An extended decoupled Kalman filter is used to update the parameters of nearest rule. Consistently under-performing rules are eventually pruned from the network.

- (c) *Sample reserve strategy*: Samples with higher information content is learnt first, and those with lower information content are pushed to the rear of the data stream to be considered for learning at a later stage in the training process. By virtue of the self-regulatory thresholds, these samples will later be considered in the learning process.

These three actions address the *what-to-learn*, *how-to-learn*, and *when-to-learn* components of meta-cognition, respectively, and helps McFIS achieve the best human learning strategy reported in literature.

The performance of McFIS is evaluated on two non-linear system identification problems discussed in [97, 169, 13], Mackey-Glass time series prediction problem [128] and Box-Jenkins gas furnace problem to estimate the output CO₂ concentration from input gas flow [63]. The performance of McFIS is compared with well-known algorithms in neuro-fuzzy framework including online sequential fuzzy extreme learning machine (OS-Fuzzy-ELM) [168], evolving Takagi- Sugeno (eTS) [12] and its variant (simpl_eTS) [13], sequential adaptive fuzzy inference system (SAFIS) [169] and recently developed algorithms like, self organizing fuzzy modified least square network (SOFMLS) [173] and fast evolving neuro-fuzzy model (ENFM) [192]. The results show the advantage of meta-cognitive neuro-fuzzy inference system over other neuro-fuzzy inference systems.

The performance of McFIS is also evaluated on a set of benchmark binary/multi-category classification problems with balanced/imbalanced data sets. The performance for classification problem is compared with SAFIS and OS-Fuzzy-ELM. The results show the advantage of meta-cognitive neuro-fuzzy inference system over other neuro-fuzzy inference systems.

3.2 Meta-Cognitive Neuro-Fuzzy Inference System

Meta-Cognitive Neuro-Fuzzy Inference System (McFIS) consists of a cognitive component and a meta-cognitive component, as shown in Fig. 3.1(b). The cognitive component

is a Takagi-Sugeno-Kang type-0 neuro-fuzzy inference system, while the meta-cognitive component is a self-regulatory learning mechanism that controls the learning ability of the neuro-fuzzy inference system. The learning mechanism is an adaptive sequential learning algorithm that starts with no rule and builds the necessary number of rules based on the information contained in the training samples. In this section, we first describe the problem or goal of the system. We will then describe the architecture and inference mechanism of the neuro-fuzzy inference system employed, followed by its self-regulatory learning mechanism.

3.2.1 Problem Definition

Let us assume that we have a stream of training data given by, $\{(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^t, \mathbf{y}^t), \dots\}$, where $\mathbf{x}^t = [x_1^t, \dots, x_m^t]^T \in \Re^m$ is the input to the network and $\mathbf{y}^t = [y_1^t, \dots, y_n^t]^T \in \Re^n$ is the corresponding target. For large range multiple-input-multiple-output dynamic system, the $\mathbf{x}^t$ could be expressed as $\mathbf{x}^t = [\mathbf{y}^{t-1}, \dots, \mathbf{y}^{t-\nu-1}; \mathbf{v}^{t-1}, \dots, \mathbf{v}^{t-\chi-1}]^T$ [12], where, $\mathbf{v}$ is the input to the dynamic system and ν and χ are the maximum lags of the output and input, respectively. During dynamical system identification, the above relationship could be expressed as:

$$\mathbf{y}^t = \mathbf{f}[\mathbf{x}^t] \quad (3.1)$$

where $\mathbf{f}[\cdot]$ is the functional relationship that maps the inputs to their respective targets ($\mathbf{x} \rightarrow \mathbf{y}$). The aim of McFIS is to approximate $\mathbf{f}[\cdot]$ such that, the predicted output

$$\hat{\mathbf{y}}^t = \hat{\mathbf{f}}[\mathbf{x}^t, \mathbf{w}] \quad (3.2)$$

is as close as possible to the desired target $\mathbf{y}^t$. It must be noted that $\mathbf{w}$ is the parameter vector of McFIS. The difference between the actual and the predicted output (error $\mathbf{e}^t = [e_1^t, \dots, e_n^t]^T$) is given by,

$$e_j^t = y_j^t - \hat{y}_j^t \quad j = 1, 2, \dots, n \quad (3.3)$$

The root mean squared prediction error is given by

$$E^t = \sqrt{\frac{1}{n} \sum_{j=1}^n (e_j^t)^2} \quad (3.4)$$

3.2.2 Cognitive Component: The Neuro-Fuzzy Inference System

In McFIS, the cognitive component is a neuro-fuzzy inference system. It is a four layer network that realizes zero order Takagi-Sugeno-Kang type fuzzy inference system [215]. The architecture of the neuro-fuzzy inference system is given in Fig. 3.2. It consists of an input layer with m nodes, a Gaussian layer with K nodes which forms the rule antecedent and the aggregation layer of an neuro-fuzzy inference system, a normalization layer with K nodes and an output layer with n nodes. The weight vector connecting normalization layer and output layer forms the rule consequent parameters of the neuro-fuzzy inference system. A detailed description of each layer is given below:

Figure 3.2: Architecture of Neuro-Fuzzy Inference System. The Inset Figure Describes the internal Structure of the i^{th} Gaussian Node.

Input Layer: The input layer is a linear layer with m nodes. It transmits the input features directly to the Gaussian layer. The output of the i^{th} input node is

$$u_i^t = x_i^t \quad i = 1, 2, \dots, m \quad (3.5)$$

Gaussian Layer: The nodes in the Gaussian layer employ Gaussian membership function to compute the membership value of each input node. The nodes in this layer

forms the antecedent and does rule aggregation operations of a fuzzy inference system. Similar to other neuro-fuzzy systems, McFIS exploits the functional equivalence between a radial basis function and a fuzzy inference system [214]. Without loss of generality let us assume that K rules are grown from $t - 1$ samples. The overall membership values of the input features are computed by each of the K rules of Gaussian layer. The firing strength of k^{th} rule is given by

$$\phi_k(\mathbf{u}^t) = \prod_{i=1}^m \exp \left(\frac{-(u_i^t - \mu_{ki})^2}{2(\sigma_k)^2} \right), \quad k = 1, 2, \dots, K \quad (3.6)$$

where, μ_{ki} is the center of the k^{th} Gaussian node for the i^{th} input feature and σ_k is the width of the k^{th} node. The above equation could be rewritten as,

$$\phi_k(\mathbf{u}^t) = \exp \left(\frac{-\|\mathbf{u}^t - \boldsymbol{\mu}_k\|^2}{2(\sigma_k)^2} \right), \quad k = 1, 2, \dots, K \quad (3.7)$$

Normalization Layer: This layer provides average firing potential for each rule. The output of the k^{th} rule is

$$\bar{\phi}_k = \frac{\phi_k}{\sum_{l=1}^K \phi_l}, \quad k = 1, 2, \dots, K \quad (3.8)$$

Output Layer: The output layer is a linear layer with n nodes. The predicted output is given by

$$\hat{y}_j^t = \sum_{k=1}^K \alpha_{jk} \bar{\phi}_k \quad j = 1, 2, \dots, n \quad (3.9)$$

where, α_{jk} is the weight connecting k^{th} normalized node and the j^{th} output node.

For a given input $\mathbf{x}^t$, McFIS calculates the predicted output $\hat{\mathbf{y}}^t$ by processing the input through all the four layers.

3.2.3 Meta-cognitive Component: The Self-Regulatory Learning Mechanism

A self-regulatory learning mechanism forms the meta-cognitive component of McFIS. The self-regulatory learning mechanism controls the learning ability of the cognitive component by deciding *what-to-learn*, *when-to-learn* and *how-to-learn*, as shown in Fig. 3.3. As a result, when a sample is presented to the neuro-fuzzy inference system, the meta-cognitive component of McFIS controls the sequential learning process by invoking one of the three strategies of *sample delete*, *sample learning* and *sample reserve*.

Figure 3.3: Schematic Diagram Highlighting the Self-regulating Learning Mechanism in McFIS.

- **Action (a) *Sample deletion strategy***- When a sample has no significant information for the network to learn, it is deleted from the training data set, without participating in the training process. Meta-cognitive aspect of *what-to-learn* is realized by this strategy.
- **Action (b) *Sample learning strategy***- The sample is used to either add/prune a rule or update the network parameters of any existing rule. The *how-to-learn* aspect of meta-cognition is achieved employing this strategy.
- **Action (c) *Sample reserve strategy***- The current sample is reserved for future use. These reserved samples will be used for fine tuning the parameters or will be discarded without participating in learning based on the self-regulatory learning thresholds. The meta-cognitive component of *when-to-learn* is realized by this strategy.

These three self-regulatory strategies of meta-cognition helps McFIS to generalize the functional relationship between the input and output efficiently. We now describe, how these strategies are implemented in McFIS.

- **(a) *Sample deletion strategy***: If the predicted error for a particular sample is below a threshold, then the knowledge in the current sample is similar to one already present in the network. As a result, it is deleted without being used in the learning process. The *sample deletion* criterion is given by

$$\text{IF } E^t \leq E_d, \quad \text{THEN delete the sample without being learnt} \quad (3.10)$$

where, E_d is the delete threshold which is chosen according to the desired prediction accuracy. The sample deletion strategy prevents the learning of similar samples and thereby avoids over-training and reduces the computational effort.

- **(b) *Sample learning strategy***: This strategy works by either ‘growing’ a new rule, ‘updating’ the parameters of the rules or pruning of insignificant rules from the network.

Rule growing criterion: When a sample contains significant novel knowledge, a new rule has to be added to the network to capture this knowledge (novelty).

Existing algorithms in neuro-fuzzy framework use error based criteria or influence of a rule as a measure of novelty. In addition to this error-based criterion, McFIS also uses the spherical potential that is widely used in kernel methods [183, 79] as a measure of novelty.

A brief description of spherical potential: The novelty is measured based on the projection of an input feature $\mathbf{x}^t$ on to a hyper-dimensional feature space $\mathbb{S}$. Since a Gaussian function is employed for projection, the hyper-dimensional feature space will be spherical in nature [79]. In McFIS, the center ($\boldsymbol{\mu}$) and the width (σ) of the Gaussian rule antecedents describe the hyper-dimensional feature space $\mathbb{S}$. Let the origin of K -dimensional space be $\phi_0 = \frac{1}{K} \sum_{k=1}^K \phi(\boldsymbol{\mu}_k)$. The spherical potential (ψ^t) of any sample ($\mathbf{x}^t$) in the feature space is expressed as the squared distance from the hyper-dimensional mapping $\mathbb{S}$ centered at origin ϕ_0 [79] and is given by

$$\psi^t = \|\phi(\mathbf{x}^t) - \phi_0\|^2 \quad (3.11)$$

From [79] the above equation can be expanded as

$$\psi^t = \phi(\mathbf{x}^t, \mathbf{x}^t) - \frac{2}{K} \sum_{k=1}^K \phi(\mathbf{x}^t, \boldsymbol{\mu}_k) + \frac{1}{K^2} \sum_{k1, k2=1}^K \phi(\boldsymbol{\mu}_{k1}, \boldsymbol{\mu}_{k2}) \quad (3.12)$$

From the eqn.(3.12), it could be seen that for Gaussian function, the first term $\phi(\mathbf{x}^t, \mathbf{x}^t)$ and the last term ($\frac{1}{K^2} \sum_{k1, k2=1}^K K \phi(\boldsymbol{\mu}_{k1}, \boldsymbol{\mu}_{k2})$) are constants and could be discarded as the potential is a measure of novelty. The spherical potential (ψ^t) is defined as the absolute value of quantity in squared distance mapping (excluding the constant term)

$$\psi^t = \left| \frac{1}{K} \sum_{k=1}^K \phi(\mathbf{x}^t, \boldsymbol{\mu}_k) \right| \quad (3.13)$$

The spherical potential is a direct measure of the knowledge contained in the sample, i.e., a higher potential (close to one) indicates that the sample is similar to the existing knowledge in the network and a lower potential (closer to 0) indicates that the sample is novel.

Therefore,

$$\text{IF } E^t > E_a \text{ AND } \psi^t < E_s \text{ THEN add a new rule} \quad (3.14)$$

where, E_S is the novelty threshold and E_a is the self-regulatory adding threshold. When a new rule is added, E_a is self-adapted according to the equation,

$$E_a = \delta E_a + (1 - \delta)E^t \quad (3.15)$$

δ being the slope that controls the contribution of prediction error in self-regulating the thresholds. For all the problems in this chapter δ is set close to 1. The self-regulatory adding threshold, E_a , is designed such that it captures the global knowledge first and at a later stage fine tunes the network by virtue of its self-regulatory nature.

When the $K + 1^{th}$ fuzzy rule is added, the antecedent parameters are initialized as

$$\begin{aligned} \boldsymbol{\mu}_{K+1} &= \mathbf{x}^t \\ \sigma_{K+1} &= \kappa * \min_{\forall j} \|\mathbf{x}^t - \boldsymbol{\mu}_j\| \quad j = 1, 2, \dots, K \end{aligned} \quad (3.16)$$

where κ is the parameter which controls the overlap between the rule antecedents. For the problems considered in this chapter, κ is chosen in the range $[0.5, 0.9]$. Lower value of κ will result in little overlap among rules which will affect the generalization ability of the network. Similarly, too much overlap among rules will also affect the performance adversely.

While initializing the output weight ($\boldsymbol{\alpha}_{K+1}$), the existing neuro-fuzzy algorithms do not fully exploit the localization property (respond to only a local region of input space [139]) of Gaussian (membership) function, efficiently [12, 13, 169]. On initialization of output weight, ($\boldsymbol{\alpha}_{K+1}$), the predicted output, $\hat{\mathbf{y}}^t$, with $K + 1$ rules should be equal to the actual output, $\mathbf{y}^t$. In various recent algorithms this condition is not fulfilled. They do not consider the contribution of the newly added rule during weight estimation. Since McFIS employs a Gaussian function as the membership function, the firing strength of the new rules for the current sample is one ($\phi_{K+1} = 1$). The output weight ($\boldsymbol{\alpha}_{K+1}$) is initialized as

$$\alpha_{j,K+1} = y_j^t - \frac{\sum_{k=1}^K \alpha_{j,k} \phi_k}{1 + \sum_{l=1}^K \phi_l}, \quad j = 1, 2, \dots, m \quad (3.17)$$

Rule parameter update criterion: The parameters of the nearest rule is updated, when the prediction error ($\|\mathbf{e}\|$) is greater than self-regulatory parameter

update threshold, E_l , i.e.,

$$\text{IF } E^t \geq E_l, \text{ THEN update rule parameters} \quad (3.18)$$

The threshold, E_l , is self-adapted based on the prediction error as:

$$E_l = \delta E_l + (1 - \delta) E^t \quad (3.19)$$

where, δ is the slope control parameter. The parameter update threshold is designed such that it captures the global knowledge first then fine tunes the network by virtue of its self-regulatory nature. For all the problems in this chapter δ is set close to 1.

The parameters of the nearest rule ($\mathbf{w}_{nr} = [\boldsymbol{\alpha}_{nr}, \boldsymbol{\mu}_{nr}, \sigma_{nr}] \in \Re^{1 \times (m+n+1)}$) are updated as

$$\mathbf{w}_{nr} = \mathbf{w}_{nr} + \mathbf{e}^t (\mathbf{G})^T \quad (3.20)$$

where, nr represents the rule antecedent which is nearest to the current input $\mathbf{x}^t$ in a Euclidean sense, $\mathbf{e}^t$ is the error defined in Eqn. (3.3) and $\mathbf{G} \in \Re^{(m+n+1) \times n}$ is the Kalman gain matrix given by

$$\mathbf{G} = \mathbf{P}_{nr} \mathbf{a} [\mathbf{R} + \mathbf{a}^T \mathbf{P}_{nr} \mathbf{a}]^{-1} \quad (3.21)$$

Here, $\mathbf{a} \in \Re^{(m+n+1) \times n}$ is the gradient of the output with respect to the parameters ($\mathbf{w}_{nr}$), $\mathbf{R} = r_0 I_{n \times n}$ is the variance of measurement noise and $\mathbf{P}_{nr} \in \Re^{(m+n+1) \times (m+n+1)}$ is the error covariance matrix of the nearest rule. Note that I is the identity matrix.

The gradient ($\mathbf{a}$) is given by

$$\mathbf{a} = \begin{bmatrix} \frac{\partial \hat{y}_1}{\partial \boldsymbol{\alpha}_{1,nr}} & \dots & \frac{\partial \hat{y}_j}{\partial \boldsymbol{\alpha}_{1,nr}} & \dots & \frac{\partial \hat{y}_n}{\partial \boldsymbol{\alpha}_{1,nr}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial \hat{y}_1}{\partial \boldsymbol{\alpha}_{n,nr}} & \dots & \frac{\partial \hat{y}_j}{\partial \boldsymbol{\alpha}_{n,nr}} & \dots & \frac{\partial \hat{y}_n}{\partial \boldsymbol{\alpha}_{n,nr}} \\ \frac{\partial \hat{y}_1}{\partial \boldsymbol{\mu}_{1,nr}} & \dots & \frac{\partial \hat{y}_j}{\partial \boldsymbol{\mu}_{1,nr}} & \dots & \frac{\partial \hat{y}_n}{\partial \boldsymbol{\mu}_{1,nr}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial \hat{y}_1}{\partial \boldsymbol{\mu}_{m,nr}} & \dots & \frac{\partial \hat{y}_j}{\partial \boldsymbol{\mu}_{m,nr}} & \dots & \frac{\partial \hat{y}_n}{\partial \boldsymbol{\mu}_{m,nr}} \\ \frac{\partial \hat{y}_1}{\partial \sigma_{nr}} & \dots & \frac{\partial \hat{y}_j}{\partial \sigma_{nr}} & \dots & \frac{\partial \hat{y}_n}{\partial \sigma_{nr}} \end{bmatrix} \quad (3.22)$$

and

$$\frac{\partial \hat{y}_j}{\partial \alpha_{i,nr}} = \bar{\phi}_{nr}, \quad j = 1, 2, \dots, n \quad (3.23)$$

$$\frac{\partial \hat{y}_j}{\partial \mu_{i,nr}} = 2\phi_{nr} \frac{x_i - \mu_{i,nr}}{\sigma_{nr}^2} \frac{\alpha_{j,nr} - \hat{y}_j}{\sum_{l=1}^K \phi_l}, \quad i = 1, 2, \dots, m \quad (3.24)$$

$$\frac{\partial \hat{y}_j}{\partial \sigma_{nr}} = 2\phi_{nr} \frac{\|\mathbf{x} - \boldsymbol{\mu}_{nr}\|^2}{\sigma_{nr}^3} \frac{\alpha_{j,nr} - \hat{y}_j}{\sum_{l=1}^K \phi_l} \quad (3.25)$$

The error covariance matrix ($\mathbf{P}_{nr}$) is updated as

$$\mathbf{P}_{nr} = [I_{z \times z} - \mathbf{G}\mathbf{a}^T] \mathbf{P}_{nr} + q_0 I_{z \times z} \quad (3.26)$$

where, $z = m + n + 1$ and the scalar quantity (q_0) determines the allowed step in the direction of gradient vector.

Rule pruning strategy: A rule is deemed to be insignificant and pruned from McFIS if its contribution (β_k) is lower than a pruning threshold, E_p , for N_w consecutive samples. If E_p and N_w are chosen very large, more rules will be pruned from the network and it may lead to system instability. Also, lower value of E_p and N_w will lead to few rules being pruned thereby increasing the network size and affecting the network performance. The choice of E_p and N_w depends on the training data size. The contribution of a rule (β_k) for a rule k is given as

$$\beta_k = \bar{\phi}_k \max_j |e_j \alpha_{kj}| \quad j = 1, \dots, n. \quad k = 1, \dots, K \quad (3.27)$$

where, ψ^t (Eqn.(3.13)) is the spherical potential calculated, ϕ_k (Eqn.(3.7)) is the firing strength of the k^{th} rule and α_k is the output weight of the k^{th} rule.

When a rule is added to the network, the dimensionality of error covariance matrix is updated as [169, 15]

$$\mathbf{P} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & p_0 I_{z \times z} \end{bmatrix} \quad (3.28)$$

where p_0 is the measure of estimated uncertainty. Similarly when k^{th} rule is pruned from the network, the dimensionality of the error covariance matrix is reduced by removing the respective rows and columns of the matrix $\mathbf{P}$, i.e., remove $(k-1)(m+n+1) + 1$ to $k(m+n+1)$ rows/columns of matrix $\mathbf{P}$.

- (c) *Sample reserve strategy*: Unlike *sample deletion strategy* that delete samples which do not contain any new information, *sample reserve strategy* defers the learning process to a later stage. This is achieved by moving the current sample to the rear-end of the training data stream. These samples may be used at a later stage in learning process by the virtue of self-regulatory nature of McFIS. The meta-cognitive component of *when-to-learn* is realized in McFIS by the *sample reserve strategy*.

Thus the *sample delete strategy*, *sample learning strategy* and *sample reserve strategy* helps McFIS actualize the meta-cognitive components of *what-to-learn*, *when-to-learn* and *how-to-learn*.

The proposed McFIS is summarized in Algorithm 2.

```
while samples in data-stream are learnt do
  for each input  $\mathbf{x}^t$  do
    Calculate the network output  $\hat{\mathbf{y}}^t$  (Eqn.(3.9)), prediction error,
     $E^t$  (Eqn.(3.4)) and significance  $\psi^t$  (Eqn.(3.13)) for a given input  $\mathbf{x}^t$ .
    if  $E^t \leq E_d$  then
      Sample is deleted from training sequence without learning.
    else if  $E^t \geq E_a$  AND  $\psi^t < E_s$  then
      A new rule is added according to eqs. (3.16), (3.17), (4.37). The
      self-regulating adding threshold is updated according to Eqn.
      (3.15).
    else if  $E^t \geq E_l$  then
      The parameters of the nearest rule is updated according to
      Eqn. (3.20). The self-adaptive learning threshold is adapted
      according to Eqn. (3.19).
    else if  $\beta_k \leq E_p$  consistently then
      Prune the rule from the network and update the error
      covariance matrix  $\mathbf{P}$ .
    else
      Push the sample to the rear-end of the sample pool to be learnt
      later. if a sample is not learnt after few processing then
        remove it from the data-stream.
      end
    end
  end
end
```

Algorithm 1: Pseudo-code of Meta-Cognitive Neuro-Fuzzy Inference System.

3.2.4 Discussion on the Influence of Self-Regulatory Thresholds

In this section, we shall discuss the influence of the thresholds employed in McFIS. There are four new parameters which needs to be initialized, delete threshold (E_d), novelty threshold (E_S), self-regulatory adding threshold (E_a) and self-regulatory parameter update threshold (E_d). In order to measure the error, we employ root mean squared error and the actual output is normalized in the range -1 to 1.

- Delete threshold (E_d): If the predicted error is less than delete threshold E_d , then the knowledge in the current sample is similar to that present in McFIS. As a result, these samples will be deleted without being learnt. Delete threshold is chosen based on the expected acceptable root mean squared error. If E_d is chosen close to zero, most of the samples will be employed for learning leading to over-fitting, whereas, E_d close to 1 will result in most of the samples being deleted without being learnt leading to under-training. The recommended range of this threshold is from [0.0001 0.001].
- Novelty threshold (E_S): This threshold measures the novelty of the current sample with respect to the network based on spherical potential. The possible range of spherical potential is between [0 1]. If spherical potential is close to 1, the current sample is similar to the knowledge contained in the network, else it is novel. As a result, if E_S is kept close to 0, no rules will be added to the network and similarly, if this threshold is kept close to 1, all the samples will be identified as novel. The suggested range for this threshold is [0.4 0.7].
- Self-regulatory add threshold (E_a): If the predicted error is higher than the add threshold, the inference system has not captured the functional relationship between input and the output efficiently. As a result a new rule will be added to the network. If the add threshold is close to 1, no new rules will be added to the network, whereas E_a close to 0 will result in frequent rule addition. For the problems in this chapter, if the prediction error is higher than [0.1, 0.5], a new rule has to be added to capture this knowledge. If the range is kept too high, few new rules will be added to capture the knowledge which will affect generalization, where as too low range will result in

large network size. The self-adaptivity of the add threshold will identify and learn the samples with significant knowledge first and then fine tunes the parameters of the rules employing samples with less information. The novelty threshold and add threshold are chosen according to the required network size and required accuracy.

- Self-regulatory parameter update threshold (E_l): The network should adapt the rule parameters based on a given sample, if the sample contains significant information which is not novel to the system. In this chapter, E_l is chosen in the range $[0.001 \ 0.01]$. If this threshold is chosen close to delete threshold E_d , no rules will be updated, whereas choosing E_l close add threshold E_a will result in frequent rule update. This might lead to over-fitting issues.

One can choose the parameters within the suggested range or can perform grid search in this range employing cross-validation techniques.

Next, we shall evaluate the performance of the proposed McFIS on standard benchmark problems.

3.3 Performance Evaluation of McFIS for Function Approximation Problems

In the previous section, the principle and working of the sequential learning algorithm for meta-cognitive neuro-fuzzy inference system was discussed in detail. In this section, the effectiveness of McFIS is demonstrated on two non-linear system identification problems discussed in [97, 169, 13], Mackey-Glass time series prediction problem [128] and Box-Jenkins gas furnace problem to estimate the output CO₂ concentration from input gas flow [63]. The performance of McFIS is compared with well-known algorithms in neuro-fuzzy framework like sequential adaptive fuzzy inference system (SAFIS) [169], evolving Takagi- Sugeno (eTS) [12] and its variant (simpl_eTS) [13], recently developed algorithms like, self organizing fuzzy modified least square network (SOFMLS) [173], on-line sequential fuzzy extreme learning machine (OS-Fuzzy-ELM) [168] and fast evolving neuro-fuzzy model (ENFM) [192].

The performance of McFIS and OS-Fuzzy-ELM was evaluated on Matlab R2011b runtime environment with 4GB memory. The parameters of McFIS was optimized employing grid search based technique and that of OS-fuzzy-ELM was optimized by employing constructive-destructive based technique. The results show the advantage of meta-cognitive neuro-fuzzy inference system over other neuro-fuzzy inference systems.

3.3.1 Performance measures

In this study, two measures are employed for evaluating the performance of the algorithm.

- Root Mean Square Error (RMSE) as used in [97, 63] is employed as a performance measure. RMSE is defined as,

$$RMSE = \sqrt{\frac{\sum_{t=1}^N (E^t)^2}{N}}$$

where, N is the total number of samples, E^t is the prediction error for the t^{th} sample as defined in Eqn.(3.4). It should be noted that, RMSE is an accuracy based performance measure.

- The percentage of samples used (PS) given by

$$PS = \frac{\text{number of samples used for training}}{\text{total number of samples available for training}} * 100\%$$

is used as an additional measure for comparison.

3.3.2 Non-linear System Identification Problem 1

First we shall consider non-linear system identification problem as described by Juang and Ling [97]. The system to be identified is given by

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \quad (3.29)$$

where, $u(t)$ is the training input and $y(t)$ is the corresponding output. The input is generated using sinusoidal function given by, $u(t) = \sin(2\pi t/100)$. The network is trained with 50,000 samples and tested with 200 samples. The output, $y(t)$ follows uniform sample distribution in the range $[-1.5, 1.5]$.

For this problem, the control parameters are chosen as follows: meta-cognitive parameters: {delete threshold, $E_d = 0.002$, add threshold, $E_a = 0.25$, parameter update threshold, $E_l = 0.12$ } novelty threshold, $E_S = 0.5$, pruning parameters: {pruning threshold, $E_p = 0.5$, pruning window, $N_w = 90$ }, EKF parameters: $\{p_0 = 1.05, q_0 = 0.007, r_0 = 1.5\}$, learning slope control parameter, $\delta = 0.006$, rule antecedent overlap constant, $\kappa = 0.9$. The logarithmic prediction error of the system trained using these parameters is given in Fig.3.4 for every 100^{th} training sample. It can be observed from Fig.3.4 that the most of the samples do not contribute to the learning process. The meta-cognitive components of *what-to-learn*, *when-to-learn* and *how-to-learn* has helped McFIS learn the underlying function well. We shall now consider how the different strategies present in meta-cognition aids accurate approximation using McFIS.

Figure 3.4: Training Logarithmic Prediction Error for Non-Linear System Identification Problem 1 on McFIS

- *Sample Deletion Strategy*: When the prediction error of a sample is less than the delete threshold, it is deleted without being used in the learning process. We shall exemplify the working of this strategy with a figure. Fig. 3.5 gives a snapshot of prediction error for samples in the range 2100-2200 along with the delete threshold (E_d). Those samples with error lower than E_d (0.002) are deleted without participating in the learning process. In Fig. 3.5, the prediction error for sample at instant 2120 is lower than the expected error given by the delete threshold and is thus removed without participating in the learning process. By deleting samples

with negligible novel information, the sample deletion strategy helps the network avoid over-training and save computational effort.

Figure 3.5: Illustration of Sample Deletion Strategy of McFIS for Non-Linear System Identification Problem 1

- *Sample Learning Strategy:* The sample learning strategy works by either growing or pruning a rule in the network or updating the antecedent and consequent parameters of the nearest rule in the network based on the information present in the sample. *Spherical potential* along with self-regulating error based criteria are employed to determine the novelty of a sample. Let us study the effect of these measures in this problem by considering a snap-shot of first 100 samples. The spherical potential for these samples and the novelty threshold are given in Fig.3.6.a, whereas, Fig.3.6.b gives the prediction error, self-regulatory adding and parameter update threshold for these 100 samples.

A sample contains significant novelty if the spherical potential calculated is less than the novelty threshold. In McFIS, a new rule is added to the network if the spherical potential calculated is lesser than the novelty threshold (0.5) and error is greater than the self-regulatory adding threshold. It could be noticed from the figures that even though the sample is novel, new rule is not added to the network if the error based criterion is not satisfied. Let us consider sample at instant 11.

Figure 3.6: Depiction of Variation in Spherical Potential, Prediction Error, and Self-Regulatory Thresholds of McFIS for System Identification Problem 1

Since both the spherical potential calculated is lower than the novelty threshold and the instantaneous error is higher than the adding threshold, a new rule is added to the network. In McFIS, in addition to spherical potential, error based criterion should also be satisfied for rule addition. One could notice from Fig.3.6.b that upon adding a new rule, the self-regulatory adding threshold adapts its value based on the prediction error.

If the prediction error of a sample is greater than the self-regulatory parameter update threshold, the sample is used for updating the parameters of the network. In the Fig.3.6.a the samples in the range 35-45 does not contain significant novel information to be considered for rule growing and are thus used to update the parameters of the nearest rule. The self-regulatory parameter update threshold

adapts its value based on the prediction error until it reaches the delete threshold.

- *Sample Reserve Strategy*: The samples which do not satisfy any of the learning criteria described above are reserved by the network to be considered for learning later. These samples, by the virtue of the self-regulatory nature of the adding and parameter update thresholds, may be used in the learning process at a later time. In the Fig.3.6.b, the samples marked with a diamond are reserved to be used later in the training process. It could be noticed that these samples do not satisfy either the adding or the parameter update criterion at the instant they were presented. These samples will be pushed to the rear of the data-stream to be learnt later. It should be noted that these samples may be used in the learning process at a later stage to fine tune the network.

We shall now perform a quantitative analysis of McFIS with existing algorithms. The result of performance analysis is given in Table 3.1. For this problem, McFIS is compared with well-known SAFIS [169], eTS [12], OS-fuzzy-ELM [168] and simpl_eTS [13] algorithms. The Table 3.1 gives the number of rules employed, training and testing RMSE and the percentage of samples used by each of the algorithms. From the table, we can observe that the McFIS outperforms other algorithms considered. The McFIS attains better results than the next best performing algorithm SAFIS by utilizing merely 5 rules and 25% of the samples. McFIS is able to attain error an order of magnitude lower than other algorithms. The meta-cognitive component of *what-to-learn* has helped McFIS remove the rest 75% of the samples, whereas the *how-to-learn* and *when-to-learn* component together with the self-regulatory learning thresholds have helped McFIS approximate the underlying function accurately.

3.3.3 System Identification Problem 2

Let us consider another non-linear system identification problem which has been studied extensively [169, 13, 173], given by:

$$y(t) = \frac{y(t-1) * y(t-2) * (y(t-1) - 0.5)}{1 + y^2(t-1) + y^2(t-2)} + u(t-1) \quad (3.30)$$

where $u(t)$ and $y(t)$ are the input and output at t^{th} instant.

Table 3.1: Performance Comparison of McFIS with Other State-of-the-art Methods for Non-linear System Identification Problem 1

Algorithm	# rules	CPU time (sec)	Train RMSE	Test RMSE	PS
SAFIS [169]	17	140	0.0539	0.0221	100%
eTS [12]	49	-	0.0292	0.0212	100%
OS-Fuzzy-ELM [168]	30	-	0.129	0.049	100%
simpl_eTS [13]	22	-	0.0528	0.0297	100%
FLEXFIS var B [125]	8	-	-	0.0171	100%
eTS+ [10]	5	-	-	0.0103	100%
DeTS [164]	4	-	-	0.0172	100%
McFIS	5	20	0.0023	0.0023	25%

As in the previous studies [169, 173], the input to the system is given by $u(t) = \sin(2\pi t/25)$ and the equilibrium state of the unforced system is (0,0). For training, 5000 data samples are generated and the trained model is tested using 200 data samples. The performance of the McFIS is compared with SAFIS [169], eTS [12], simpl_eTS [13] and SOFMLS [173]. For this study, the network parameters are chosen as: $\{E_d = 0.002, E_a = 0.35, E_l = 0.08\}$, $E_S = 0.15$, $\{E_p = 0.07, N_w = 14\}$, $\{p_0 = 1.9, q_0 = 0.03, q_0 = 0.03, r_0 = 1.5\}$, $\delta = 0.03$, $\kappa = 0.56$. The performance comparison of McFIS using the above parameters is given in Table 3.2. Table 3.2 gives the number of rules, training and testing RMSE and percentage of samples used by each of the algorithms.

Table 3.2: Performance Comparison of McFIS with Other State-of-the-art Methods for Non-linear System Identification Problem 2

Algorithm	# rules	CPU time (sec)	Train RMSE	Test RMSE	PS
SAFIS [169]	17	4.2	0.0539	0.0221	100%
eTS [12]	49	-	0.0292	0.0212	100%
simpl_eTS [13]	22	-	0.0528	0.0225	100%
SOFMLS [173]	5	-	0.0341	0.0201	100%
McFIS	5	2.3	0.002	0.004	51%

From Table 3.2 it could be seen that McFIS using the same number of rules as next best performing algorithm SOFMLS, attains a network which produces significantly lower error. McFIS attains this performance by utilizing merely 50% of the training data. The spherical potential as a measure of novelty, is able to select samples which contains new knowledge and the complete exploitation of the Gaussian rule has helped the network

approximate the samples well. In addition, the learning strategies has helped McFIS in choosing what samples to be learnt, when and how to learn those samples.

3.3.4 Mackey-Glass Time Series Problem

One of the classical benchmark problem in literature is the Mackey-Glass time series prediction [128, 42]. This chaotic time series data is generated from the differential equation:

$$\frac{dx}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \quad (3.31)$$

where, $\tau = 17$ and $x(0) = 1.2$. Aim of this study is to predict $x(t + \hat{h})$ from $[x(t), x(t - \Delta t), \dots, x(t - (n-1)\Delta t)]$. For the sake of this study, we consider the following conditions: $n = 4$, $\Delta t = 6$, $\hat{h} = 85$ and the numerical solution to the differential Eqn.(3.31) is found using the forth order Runge-Kutta method.

For comparing the performance of Mackey-Glass series data, we employ a normalized root mean squared error called Non-Destructive Error Index (NDEI) [169, 13]. NDEI is defined as RMSE 3.3.1 divided by the standard deviation, S.D, of the target data:

$$NDEI = \frac{RMSE}{S.D}.$$

Table 3.3: Performance Comparison of McFIS with Other State-of-the-art Methods for Mackey-Glass Stochastic Time Series Problem

Algorithm	# rules	CPU time (sec)	Test NDEI
eTS [12]	99	0.37	0.356
Simpl_eTS [13]	21	-	0.376
SAFIS [169]	21	-	0.38
ENFM [192]	8	-	0.06
McFIS	6	0.33	0.03

Table 3.3 gives the performance comparison of Mackey-Glass time series prediction with well-known algorithms like eTS [12],simpl_eTS [13], SAFIS [169] and ENFM [192]. Table 3.3 gives the number of rules employed and testing NDEI for each algorithm employed. To generate this result the following parameter setting was employed: $\{E_d = 0.0004, E_a = 0.4, E_l = 0.006\}$, $E_S = 0.4$, $\{E_p = 0.02, N_w = 37\}$, $\{p_0 = 1.4, q_0 = 0.003, r_0 = 1.1\}$, $\delta = 0.004$, $\kappa = 0.6$. It could be seen that McFIS attains the best performance

with the least number of rules. It should be noted that McFIS uses about 98% of the samples to learn the underlying function in Mackey-Glass time series prediction problem. From the results, we can see that the proposed McFIS approximates the decision surface more efficiently than existing results in the literature.

3.3.5 Box-Jenkins Gas Furnace Problem

The system identification of Box-Jenkins gas furnace [63] is a well-known problem. The aim of the problem is to predict the output CO₂ concentration from input gas flow rate in a furnace. The data set consists of 290 data samples of which, 200 samples are used for training and rest 90 samples are used for testing. For McFIS, a series-parallel model, $\hat{y}(t) = f(y(t-1), u(t-4))$ is used to model the system.

Table 3.4: Performance Comparison of McFIS with Other State-of-the-art Methods for Box-Jenkins Gas Furnace Problem

Algorithm	# rules	CPU time (sec)	Test RMSE
SAFIS [169]	5	0.26	0.071
eTS [12]	5	-	0.049
simpl_eTS [13]	3	-	0.0485
SOFNN [97]	4	-	0.48
SOFMLS [173]	5	-	0.047
McFIS	4	0.23	0.036

The performance of McFIS is compared with other algorithms in neuro-fuzzy framework like, eTS [12], simpl_eTS [13], SOFNN [114] and SOFMLS [173]. The following thresholds were utilized to generate the result in this chapter: $\{E_d = 0.0003, E_a = 0.44, E_l = 0.01\}$, $E_S = 0.3$, $\{E_p = 0.07, N_w = 78\}$, $\{p_0 = 1.2, q_0 = 0.001, r_0 = 1.2\}$, $\delta = 0.004$, $\kappa = 0.8$. It could be observed from Table 3.4 that, McFIS approximates the functions better than other algorithms using fewer rules. Although SOFNN uses the same number of rules as McFIS, McFIS achieves significantly better performance. For Box-Jenkins Gas Furnace Problem, McFIS utilizes about 98% of the samples to learn the underlying data distribution.

3.4 Discussion on Effects of Meta-Cognition

In this chapter two significant contributions are made: (a) exploitation of localization property of Gaussian rule during rule addition and (b) meta-cognitive learning philosophy for training an NFIS. Authors argue that with the inclusion of these principles, the performance of any algorithm could be improved. In order to buttress these arguments, three separate experiments are performed using SAFIS algorithm.

- *Experiment A*: When a growing criterion is satisfied in SAFIS, the output weight (α_{K+1}) is updated using Eq. (3.17). The above changes will help SAFIS to exploit the localization property of the Gaussian rules effectively. We call this algorithm as ‘SAFIS*’.
- *Experiment B*: The ‘SAFIS*’ is trained using the samples selected by McFIS to approximate the function. The selected training samples are presented one-by-one in the same order in which they are used in the McFIS learning algorithm. We call this algorithm as ‘SAFIS[†]’.
- *Experiment C*: The ‘SAFIS*’ is trained using the selected samples by McFIS followed by the deleted samples. We call this algorithm as ‘SAFIS[‡]’.

The three experiments are validated on system identification problem 1 and system identification problem 2. The results are available in Table 3.5.

From the table it could be noticed that by exploiting the localization property of Gaussian rule, there is a significant improvement in performance of SAFIS* compared to SAFIS, i.e., with fewer rules the network is better able to capture the knowledge in the samples. Earlier we had argued that presenting similar samples to the network lead to over-training and *what-to-learn* strategy has to be included in the learning algorithm. In order to support the claim, ‘Experiment B’ is conducted. The results from Table 3.5 indicates SAFIS[†], employing fewer samples (selected by *what-to-learn* and *when-to-learn* components of McFIS), is able to attain better training and testing performance in comparison to SAFIS and SAFIS*. ‘Experiment C’ which employs samples selected by McFIS followed by deleted samples clearly shows the over-training in both the cases. The difference in the performance of SAFIS-variants and McFIS might be due to the difference in *how-to-learn* strategy in both the algorithms.

Table 3.5: Quantitative Verification of McFIS Learning Philosophy in SAFIS Algorithm [169] Using System Identification Problem 1

Data set	Algorithm	No. of rules	Train RMSE	Test RMSE
System Identification 1	SAFIS	17	0.0539	0.0221
	SAFIS*	12	0.0327	0.0201
	SAFIS [†]	11	0.0301	0.0171
	SAFIS [‡]	11	0.022	0.0191
	McFIS	5	0.0023	0.0023
System Identification 2	SAFIS	17	0.0539	0.0221
	SAFIS*	10	0.0152	0.0155
	SAFIS [†]	8	0.0118	0.0121
	SAFIS [‡]	8	0.0115	0.0124
	McFIS	5	0.002	0.004

3.5 Performance Evaluation of McFIS for Classification Problems

In this section, the effectiveness of McFIS is demonstrated on a set of benchmark classification problems from UCI Machine Learning Repository [26]. The performance of the proposed algorithm is compared with SAFIS [169], and OS-Fuzzy-ELM [168]. The results for SAFIS, OS-Fuzzy-ELM and McFIS have been generated on a Matlab R2011b runtime environment with 4Gb memory. The parameters of SAFIS and McFIS have been optimized employing grid search based technique, whereas, OS-Fuzzy-ELM has been optimized by constructive-destructive based technique. The data sets considered along with their descriptions, including the number of input features, the number of training and testing samples, and the imbalance factor of the data set is given in Table 3.6. The imbalance factor of a data set as defined in [209] is

$$\text{I.F.} = 1 - \frac{n}{N} \min_{j=1, \dots, n} N_j \quad (3.32)$$

where N_j is the number of samples belonging to class l and $N = \sum_{j=1}^n N_j$.

3.5.1 Performance measures

In this chapter, we have employed two measures to evaluate the performance of the algorithm on classification problems.

Table 3.6: Description of Benchmark Data Sets Employed in Chapter 3

Type of Data set	Problem	# features	# classes	Number of samples		Imbalance Factor	
				Train	Test	Train	Test
Multi category	IS	19	7	210	2100	0	0
	Iris	18	4	424	422	0	0
	Wine	13	3	60	118	0	0.29
	GI	9	6	109	105	0.68	0.77
Binary	LD	6	2	200	145	0.17	0.14
	PIMA	8	2	400	368	0.22	0.39
	BC	9	2	300	383	0.26	0.33
	ION	34	2	100	251	0.28	0.28

- Overall accuracy (η_o): It is defined as

$$\eta_o = \frac{1}{N} \sum_{j=1}^n q_{jj} \times 100\% \quad (3.33)$$

where q_{jj} is the number of samples correctly classified in class j .

- Average accuracy (η_a): It is given by

$$\eta_a = \frac{1}{N} \sum_{j=1}^n \frac{q_{jj}}{N_j} \times 100\% \quad (3.34)$$

3.5.2 Performance comparison

The number of rules, overall and average testing efficiency for SAFIS, OS-Fuzzy-ELM and McFIS are reported in Table 3.7. The table contains results of both the binary and multi category classification data sets from the UCI Machine Learning Repository. From table 3.7, we can see that McFIS performs better than SAFIS and OS-Fuzzy-ELM on all the eight data sets.

On the balanced classification problems (Image Segmentation (IS) and Iris), the generalization performance of McFIS has improved by 1-5% compared to SAFIS and OS-Fuzzy-ELM. For these problems, the meta-cognitive components of *what-to-learn*, *when-to-learn* and *how-to-learn* has helped to build an efficient classifier. On the imbalanced data sets (wine, Glass Identification (GI), Liver Disorder (LD), PIMA, Breast Cancer (BC) and Ionosphere (ION)), the generalization performance of McFIS is approximately

Table 3.7: Performance Comparison of McFIS with Other State-of-the-art Methods on Classification Problems

Data sets	SAFIS			OS-Fuzzy-ELM			McFIS		
	K	Test		K	Test		K	Test	
		η_o	η_a		η_o	η_a		η_o	η_a
IS	156	90.38	90.38	30	87.33	87.33	78	91.33	91.33
Iris	24	92.38	92.38	25	97.14	97.14	6	97.14	97.14
Wine	35	95.76	95.76	30	94.07	94.07	12	96.78	96.78
GI	134	73.24	83.32	150	76.19	78.1	100	82.9	85.7
Liver	133	68.28	68.82	100	66.21	65.68	66	71.7	73.4
PIMA	179	75.82	71.46	100	68.75	66.36	133	75.81	75.65
BC	55	97.39	97.84	90	91.64	89.45	33	97.39	97.85
ION	46	91.63	91.78	50	77.69	71.34	23	94.2	93.29

1-5% more than SAFIS and OS-Fuzzy-ELM with fewer rules. The complete exploitation of Gaussian rule along with spherical potential as measure of novelty has helped McFIS to efficiently induct rules.

It could be observed from the table that although McFIS performs better than other algorithms, the performance improvement is not significantly better. It has been shown that class-specific knowledge is important for classification problems. The use of class-specific as well as hinge-loss based learning approach might help the algorithm achieve better performance.

3.6 Summary

In this chapter, a neuro-fuzzy inference system and its meta-cognitive sequential learning algorithm together referred to as Meta-Cognitive Neuro-Fuzzy Inference System (McFIS) is presented. McFIS consists of two components: a cognitive component and a meta-cognitive component. A zero order Takagi-Sugeno-Kang fuzzy inference system forms the cognitive component, while a self-regulatory sequential learning algorithm is its meta-cognitive component. As and when a samples is presented, the self-regulatory learning mechanism helps the neuro-fuzzy inference system to choose *what-to-learn*, *when-to-learn* and *how-to-learn* by activating one of the following strategies: *sample deletion strategy*, *sample learning strategy* or *sample reserve strategy*. In McFIS, the rule growing strategy employs spherical potential as a measure of novelty, and addition of a new rule exploits the

localization property of Gaussian rule, effectively. The performance of McFIS is evaluated on a set of system identification and time series prediction problems. The performance is also evaluated on a set of benchmark classification problems. The performance analysis indicate the superior performance of McFIS.

In the next chapter, the developed McFIS is modified to address classification problems. McFIS based classifier will be developed by considering class-specific information in learning.

Chapter 4

A Meta-Cognitive Sequential Learning Algorithm for Neuro-Fuzzy Inference System based Classifier (McFIS-Classifier)

4.1 Introduction

In the previous chapter 3, a meta-cognitive sequential learning algorithm for a neuro-fuzzy Inference System was proposed, which employs the principles of human meta-cognition for efficiently learning and generalizing a given system. The meta-cognitive learning mechanism helps the neuro-fuzzy inference system achieve better generalization by efficiently controlling *what-to-learn*, *when-to-learn*, and *how-to-learn*. Moreover, the complete exploitation of localization property of the Gaussian rule, along with the use of spherical potential has helped the network approximate the underlying functional relationship between input and output efficiently, by employing a compact network structure. The performance improvement of McFIS has been verified using a set of benchmark regression, time series as well as classification problems.

In case of classification problems, it has been shown in literature that class-specific information affects the performance of a classifier significantly [209]. In addition, minimizing misclassification error also plays an important role in improving classification accuracy. In the literature of neuro-fuzzy inference systems reviewed, no algorithm till date addresses these requirements of a classifier, efficiently.

In view of the above mentioned observations, in this chapter, we extend the proposed McFIS to specifically address classification problems (McFIS-classifier). The work done in this chapter has been published as an article in [204]. In order to address classification problems, class-specific criterion has been introduced in both monitory as well as control signals. To measure error, we employ hinge-loss function [257], as it can estimate posterior probability more accurately than mean squared function [210, 257]. The measure of novelty, *spherical potential* (Eqn. (3.13), page 52), has been redefined to class-specific spherical potential. Moreover, the control signals which decide *what-to-learn*, *when-to-learn*, and *how-to-learn* have also been modified to consider class-specific information.

In the cognitive component, generally when a new rule is added to the network, width of the Gaussian rule is initialized based on the distance to the nearest rule. While developing a classifier, it has been shown in [209] that such width initialization affects the performance of the network adversely. The width allocation for a new Gaussian rule has to consider distance to the nearest rule belonging to the same class and to that of other classes. In the proposed McFIS-classifier, the width of a new rule is initialized considering separation to nearest rule belonging to same class (intra-class) and to that of other classes (inter-class). Similarly, while initializing the output weights, the learning algorithm should penalize the inference system for letting the wrong class get close to correct output. Whereas, as long as there is no misclassification, there should be no penalty. This aspect of weight initialization has been introduced by employing hinge-loss based criterion.

In the next section, a McFIS-classifier is developed based on the concepts of human meta-cognition. The performance of McFIS-classifier is evaluated on a set of benchmark binary/ multi-class classification problems. Performance comparison, quantitatively as well as qualitatively, against a set of well-known neuro-fuzzy inference system and classifiers such as online sequential fuzzy extreme learning machines (OS-Fuzzy-ELM) [168], sequential adaptive fuzzy inference system (SAFIS) [169], evolving fuzzy classifier (eClass) [16], and support vector machines (SVM) [39] indicate the superior performance of McFIS-classifier.

Figure 4.1: Nelson and Narens Model of Meta-Cognition Adapted from [145] and its Analogy in McFIS-Classifier

4.2 Meta-Cognitive Neuro-Fuzzy Inference System based Classifier

Meta-cognitive neuro-fuzzy inference system based classifier (McFIS-classifier) consists of a cognitive component as well as a meta-cognitive component, as depicted in Fig. 4.1(a). Similar to McFIS proposed in chapter 3, the cognitive component is a Takagi-Sugeno-Kang fuzzy inference system, while the meta-cognitive component is a self-regulatory learning mechanism which controls the learning in the cognitive component, as shown in Fig. 4.1(b). McFIS-classifier is an adaptive sequential learning algorithm that starts with no rule and builds the necessary number of rules based on the information contained in the training data stream.

4.2.1 Problem Definition

Let us assume that we have a stream of training data $\{(\mathbf{x}^1, c^1), \dots, (\mathbf{x}^t, c^t), \dots\}$, where $\mathbf{x}^t = [x_1^t, \dots, x_m^t]^T \in \mathbb{R}^m$ is a m -dimensional input vector of t -th sample, $c^t \in [1, 2, \dots, n]$ is its class label, and n represents the total number of distinct classes.

For solving the classification problem in a neuro-fuzzy framework, the class label (c^t) is converted into a coded class label ($\mathbf{y}^t = [y_1^t, \dots, y_j^t, \dots, y_n^t] \in \mathbb{R}^n$) as shown below

$$y_j^t = \begin{cases} 1 & \text{If } j == c^t \\ -1 & \text{otherwise} \end{cases} \quad j = 1, 2, \dots, n. \quad (4.1)$$

The input space $\mathbf{x}$ provides the necessary information on the probability distribution over the observation data to predict the corresponding class labels with a certain accuracy. For this purpose, one needs to estimate the functional relationship (the decision function $f : \mathbf{x} \rightarrow \mathbf{y}$) between the input vector ($\mathbf{x}^t$) and the coded class labels ($\mathbf{y}^t$) from the stream of the training data. In this chapter, we use a neuro-fuzzy inference system with a Gaussian membership function to approximate the decision function defined by the training samples. In a sequential learning framework, the training data arrives one-by-one and is discarded after learning.

The predicted output ($\hat{\mathbf{y}}^t$) using McFIS-classifier is given by

$$\hat{\mathbf{y}}^t = f(\mathbf{x}^t, \mathbf{w}) \quad (4.2)$$

where the vector $\mathbf{w}$ are the parameters of McFIS-classifier. The difference between the actual and the predicted output (error $\mathbf{e}^t = [e_1^t, \dots, e_n^t]^T$) is defined as

$$e_j^t = \hat{y}_j^t - y_j^t, \quad j = 1, 2, \dots, n \quad (4.3)$$

The maximum absolute prediction error is given by

$$E^t = \max |\mathbf{e}^t| \quad (4.4)$$

4.2.2 Cognitive Component: The Neuro-Fuzzy Inference System

The cognitive component of McFIS-classifier is a neuro-fuzzy inference system as shown in Fig. 4.2. It is a four layer network which realizes a Takagi-Sugeno-Kang fuzzy inference system. Here, the first layer is an input layer with m nodes, the second layer is referred to as the Gaussian layer with K nodes, the third layer is referred to as a normalization layer consisting of K nodes and the fourth layer is the output layer with n nodes. A detailed description of each layer is given below:

Input Layer: The input layer has m nodes, where each input node represents an input feature. The outputs ($\mathbf{u}^t$) from the input layer are directly transmitted to the Gaussian layer. The output of the i th input node (u_i^t) is

$$u_i^t = x_i^t, \quad i = 1, 2, \dots, m \quad (4.5)$$

Figure 4.2: Architecture of Neuro-fuzzy Inference System. The inset Figure Describes the Internal Structure of the i^{th} Gaussian Node.

Gaussian Layer: Without loss of generality, we assume that the McFIS-classifier builds K rules from $t - 1$ training samples. The Gaussian layer contains the rule antecedents of each of these K rules and performs the rule inference using a t -norm operation to compute the overall contribution of the rule to the input features. The membership of the k th rule is given by

$$\phi_k(\mathbf{u}^t) = \prod_{i=1}^m \exp \left(\frac{-(u_i^t - \mu_{ki}^l)^2}{2(\sigma_k^l)^2} \right), \quad k = 1, 2, \dots, K \quad (4.6)$$

where μ_{ki}^l is the center of the k th Gaussian node for the i th input feature of the t -th sample and σ_k^l is the width of the k th node. The superscript l indicates the associated class label of the rule.

The above equation can be reduced to

$$\phi_k(\mathbf{u}^t) = \exp \left(\frac{-\|\mathbf{u}^t - \boldsymbol{\mu}_k^l\|^2}{2(\sigma_k^l)^2} \right), \quad k = 1, 2, \dots, K \quad (4.7)$$

Normalization Layer: The number of nodes in this layer is the same as in the Gaussian layer. The output of the i th normalized node is given by

$$\bar{\phi}_k = \frac{\phi_k}{\sum_{j=1}^K \phi_j}, \quad k = 1, 2, \dots, K \quad (4.8)$$

Output Layer: The number of nodes in this layer is equal to the number of distinct classes (n). The predicted output ($\hat{\mathbf{y}}^t$) is the weighted sum of the normalized output and is given by

$$\hat{y}_j^t = \sum_{k=1}^K \alpha_{jk} \bar{\phi}_k, \quad j = 1, 2, \dots, n \quad (4.9)$$

where α_{jk} is the weight connecting the k th normalized node and the j th output node.

Using the predicted output ($\hat{\mathbf{y}}^t$), the predicted class label ($\hat{c}^t$) is obtained as shown below.

$$\hat{c}^t = \arg \max_{j=1, \dots, n} (\hat{y}_j^t) \quad (4.10)$$

For a given input $\mathbf{x}^t$, McFIS-classifier predicts the corresponding class by processing the input through all the four layers.

4.2.3 Meta-Cognitive Component: The Self-Regulatory Learning Mechanism

The objective of the sequential learning algorithm is to determine the number of rules and their associated parameters to approximate the decision function as closely as possible to the original system. The learning process in cognitive component is controlled by the meta-cognitive component (self-regulatory learning mechanism). The meta-cognitive component selects the learning strategies based on monitory signals from the neuro-fuzzy inference system. In this chapter, we use sample error based measure and knowledge in current sample as monitory signal. These monitory signals are used to develop three learning strategies: *sample deletion*, *sample learning* and *sample reserve*. We discuss the monitoring and control signals in details below.

Monitory signals:

For monitory signals, we use a hinge-loss error function [257] as a measure of sample error and the newly developed class-specific spherical, potential derived using novelty measure in [79], as knowledge based measure.

- *Hinge-loss error function:* In general, mean square error is used as a measure of sample error in most of the classification problems. For a classification problem, it has been proven theoretically in [210, 257, 34] that a classifier developed using a

hinge-loss function estimates the posterior probability more accurately than a mean square error function. Hence, we use hinge-loss error ($\mathbf{e}^t = [e_1^t, \dots, e_n^t]^T \in \mathbb{R}^n$) defined below as a measure of sample error.

$$e_j^t = \begin{cases} 0 & \text{if } \hat{y}_j^t y_j^t > 1 \\ \hat{y}_j^t - y_j^t & \text{otherwise} \end{cases} \quad j = 1, 2, \dots, n \quad (4.11)$$

The hinge-loss error allows the network output to grow beyond ± 1 . Hence, truncated output of the classifier is an accurate estimate of the posterior probability [209]. It is given as

$$T(\hat{y}_j^t) = \min(\max(\hat{y}_j^t, -1), 1) \quad j = 1, 2, \dots, n. \quad (4.12)$$

Since the truncated output is in the range $[-1, 1]$, the posterior probability of input $\mathbf{x}^t$ belonging to class c is given as

$$\hat{p}(c|\mathbf{x}^t) = \frac{T(\hat{y}_j^t) + 1}{2}, \quad c = 1, 2, \dots, n. \quad (4.13)$$

The maximum absolute hinge-loss error (E^t) is defined as,

$$E^t = \max_{j=1, \dots, n} |e_j^t| \quad (4.14)$$

- *Class-Specific Spherical potential*: Spherical potential is a measure often used in kernel methods to determine the novelty of the data [79]. Here, the novelty is determined based on the projection of an input feature ($\mathbf{x}^t$) on to a hyper-dimensional feature space $\mathbb{S}$, i.e., $\mathbf{x}^t \rightarrow \phi$. Since we are using Gaussian functions for projection, the hyper-dimensional feature space will be spherical in nature [183]. In McFIS, the center ($\boldsymbol{\mu}$) and width (σ) of the Gaussian rules describe the hyper-dimensional feature space $\mathbb{S}$. Let the origin of the K -dimensional space be $\phi_0 = \frac{1}{K} \sum_{i=1}^K \phi(\boldsymbol{\mu}_i^l)$. The spherical potential (ψ) of any sample ($\mathbf{x}^t$) in the feature space is expressed as a squared distance from the hyper-dimensional mapping $\mathbb{S}$ centered at origin ϕ_0 and is given by

$$\psi^t = \|\phi(\mathbf{x}^t) - \phi_0\|^2 \quad (4.15)$$

Using the steps shown in [79], the above equation can be expanded as

$$\psi^t = \phi(\mathbf{x}^t, \mathbf{x}^t) - \frac{2}{K} \sum_{k=1}^K \phi(\mathbf{x}^t, \boldsymbol{\mu}_k) + \frac{1}{K^2} \sum_{k1, k2=1}^K \phi(\boldsymbol{\mu}_{k1}, \boldsymbol{\mu}_{k2}) \quad (4.16)$$

We can see from the above equations that the first and last terms are constants for the Gaussian function. Since the potential is a measure of novelty, these constants may be discarded and the spherical potential can be redefined as

$$\psi^t = \left| \frac{1}{K} \sum_{k=1}^K \phi(\mathbf{x}^t, \boldsymbol{\mu}_k) \right| \quad (4.17)$$

Since we are addressing the classification problems, the class-specific distribution plays a vital role and it influences the performance the classifier significantly [210]. Hence, we use the measure of the class-specific spherical potential which may be defined as the spherical potential of the current sample of class c with respect to the rules associated with the same class (i.e., $l = c$). Let K^c be the number of rules associated with the class c then the class-specific spherical potential (ψ_c) is defined as

$$\psi_c^t = \frac{1}{K^c} \sum_{k=1}^{K^c} \phi(\mathbf{x}^t, \boldsymbol{\mu}_k^c) \quad (4.18)$$

The spherical potential directly indicates the knowledge contained in the sample, i.e., a smaller potential (close to zero) indicates that the sample contains novelty and a higher potential (close to one) indicates that the sample is similar to the existing knowledge in the network. The meta-cognitive learning algorithm uses the above class-specific spherical potential for adding rules.

The meta-cognitive learning algorithm uses these monitory signals to control the learning of the neuro-fuzzy inference system. We shall discuss each of the control strategies in detail now.

Control Signals: McFIS-classifier controls the learning of a neuro-fuzzy inference system by deciding on *what-to-learn*, *how-to-learn* and *when-to-learn*. These actions are realized by *sample deletion strategy*, *sample learning strategy* and *sample reserve strategy*, respectively.

- *Sample deletion strategy*: It prevents learning similar samples repeatedly, thereby avoiding over-training and reducing the computational effort. If the predicted class label is the same as the true class label and the estimated posterior probability for that class is above a threshold (E_d), then the current sample is similar to the knowledge already present in McFIS-classifier. Hence, it is deleted without being used in the learning process. The *sample deletion* criterion is given by

$$c^t == \hat{c}^t \text{ AND } \hat{p}(\hat{c}^t | \mathbf{x}^t) \geq E_d \quad \text{THEN delete the sample without being learnt} \quad (4.19)$$

The constant E_d is related to the expected confidence level in the class identification, i.e., if McFIS-classifier predicts the class label ($\hat{c}$) of a sample with a high posterior probability (say, 0.9) then the sample contains information similar to the knowledge present in McFIS-classifier. Hence, such samples can be deleted without being used in the learning process. Details on selection of thresholds are provided at a later stage.

- *Sample learning strategy* addresses the *how-to-learn* aspect of meta-cognition and this involves allocating a new rule ('rule growing') or updating the parameters of the rules ('parameter update') or removing redundant rules ('rule pruning').

Rule Growing Criterion: When a new sample contains significant information or when the estimated class label is incorrect, then one needs to add a new rule to capture the new knowledge (novelty) in the data. Hence, proposed McFIS-classifier combine sample error and class-specific spherical potential to develop a rule growing criterion and it is defined as

$$E^t > E_a \text{ AND } c^t \neq \hat{c}^t \text{ AND } \psi_c < E_S \quad \text{THEN add a new rule} \quad (4.20)$$

where E_S is a class-specific novelty threshold and E_a is a self-adaptive adding threshold. E_a is adapted by the following equation as given below

$$E_a := \delta (E^t) + (1 - \delta)E_a \quad (4.21)$$

where δ is the slope which controls the rate of self-adaptation.

Next, we describe how the parameters of the new rule are initialized such that McFIS-classifier capture the new knowledge present in the current sample, accurately.

In classification problems, when a new rule is added, it may overlap with rule belonging to other classes resulting in misclassification. Existing neuro-fuzzy sequential learning algorithms in the literature do not consider the overlapping condition for the new rule parameter initialization. In addition, in the existing algorithms, the parameters of the new rules added do not fully exploit the localization property (respond to only a local region of input space [139]) of the Gaussian (membership) functions efficiently. Hence, in this work, the parameters of the new rule ($K + 1$) are initialized such that McFIS-classifier exploits both the overlapping condition and the localization property, effectively.

The center ($\boldsymbol{\mu}_{K+1}$), width (σ_{K+1}) and output weight ($\boldsymbol{\alpha}_{K+1}$) are initialized as follows. The center of new rule is initialized to the current sample ($\boldsymbol{\mu}_{K+1} = \mathbf{x}^t$). The width (σ_{K+1}) depends on inter-class and intra-class distance between new rule center. Here, McFIS-classifier employs nearest rule in the sample class ('intra-class') and other class ('inter-class') to estimate the width of new rule.

Let nrS be the nearest rule in the intra-class ('same class') and nrI be the nearest rule in the inter-class ('other classes'). They are defined as

$$nrS = \arg \min_{l=c; \forall k} \|\mathbf{x}^t - \boldsymbol{\mu}_k^l\|; \quad nrI = \arg \min_{l \neq c; \forall k} \|\mathbf{x}^t - \boldsymbol{\mu}_k^l\| \quad (4.22)$$

Let the Euclidian distance between the nearest intra/intra rules be defined as

$$d_S = \|\mathbf{x}^t - \boldsymbol{\mu}_{nrS}\|; \quad d_I = \|\mathbf{x}^t - \boldsymbol{\mu}_{nrI}\| \quad (4.23)$$

The distance d_S and d_I are used to identify the overlapping condition of the current sample with respect to the existing rules from different classes. The ratio ($\frac{d_S}{d_I}$) defines the closeness of the current sample with respect to the rules. If the sample is closer to the nearest rule in the same class (nrS) then the ratio $\frac{d_S}{d_I} < 1$, i.e., no overlapping condition - current sample from the class c does not fall in the region of overlap with other classes. Otherwise, $\frac{d_S}{d_I} \geq 1$ means the current sample from the class c falls in the region of overlap with the other classes. Based on the inter/intra class distance ratio, McFIS-classifier initializes the width (σ_{K+1}^l) of the new rule as:

- *no overlapping*: When the current sample ($\mathbf{x}^t$) is closer to the nearest rule in the same class, then the sample does not overlap with other classes, i.e., if the ratio $\frac{d_S}{d_I} < 1.0$ then the current sample does not overlap with the other classes. In this case, the width is initialized as

$$\sigma_{K+1}^l = \kappa * ||\mathbf{x}^t - \boldsymbol{\mu}_{nrS}|| \quad (4.24)$$

where κ determines the overlap between the new rule and nearest rule in the same class, and κ usually lies in the range $[0.5, 1]$, to facilitate better generalization among rules of same class.

- *Overlapping*: When the current sample ($\mathbf{x}^t$) is closer to the nearest rule in the inter-class, then the sample overlaps with other classes, i.e., if the ratio $\frac{d_S}{d_I} \geq 1.0$ then the current sample is overlapping with other classes and hence misclassification may occur. In this case, the width of the new rule is initialized as

$$\sigma_{K+1}^l = \eta * ||\mathbf{x}^t - \boldsymbol{\mu}_{nrI}|| \quad (4.25)$$

where η determines the overlap between the new rule and the nearest rule in the inter class. To minimize the misclassification error, the range for η is chosen as $[0.1, 0.5]$. General philosophy to adhere is $0 < \eta < \kappa \leq 1$.

Another important parameter is the output weight ($\boldsymbol{\alpha}_{K+1}$). It should be initialized such that the predicted output ($\hat{\mathbf{y}}^t$) with $K + 1$ rules is equal to the actual output ($\mathbf{y}^t$). Note that the rules employs a Gaussian function and hence the firing strength of the new rule for the current sample is one ($\phi_{K+1} = 1$). Now, by substituting the hinge loss error ($\mathbf{e}^t$) to zero in eq. (4.11), the output weight ($\boldsymbol{\alpha}_{K+1}$) can be chosen as

$$\alpha_{j,K+1} = \begin{cases} 0 & y_j^t * \frac{\sum_{i=1}^K \alpha_{ji} \phi_i}{1 + \sum_{i=1}^K \phi_i} > 1 \\ y_j^t - \frac{\sum_{i=1}^K \alpha_{ji} \phi_i}{1 + \sum_{i=1}^K \phi_i} & \text{otherwise} \end{cases} \quad j = 1, 2, \dots, n \quad (4.26)$$

Rule Parameter Update Criterion: When the maximum hinge error is greater than the self-adaptive threshold (E_l) and the predicted class label is the same as the actual class label then the parameters of the nearest rule in the same class are

updated using an extended Kalman filter algorithm. The update criterion is given by

$$E^t \geq E_l \text{ AND } c^t == \hat{c}^t \text{ THEN update rule parameters} \quad (4.27)$$

where, E_l is the self-adaptive parameter update threshold. This threshold E_l is regulated based on the prediction error as:

$$E_l := \delta E^t + (1 - \delta) E_l \quad (4.28)$$

where, δ is the slope that controls the rate of adaptation.

McFIS-classifier updates the parameters of the nearest rule in the same class ($\mathbf{w}_{nrS} = [\boldsymbol{\alpha}_{nrS}, \boldsymbol{\mu}_{nrS}^l, \sigma_{nrS}^l] \in \Re^{1 \times (m+n+1)}$) for the current sample as given below

$$\mathbf{w}_{nrS} = \mathbf{w}_{nrS} + (\mathbf{e}^t) \mathbf{G}^T \quad (4.29)$$

where, $\mathbf{e}$ is the error defined in Eq.(4.11) and $\mathbf{G} \in \Re^{(m+n+1) \times n}$ is the Kalman gain matrix given by

$$\mathbf{G} = \mathbf{P}_{nrS} \mathbf{a} [\mathbf{R} + \mathbf{a}^T \mathbf{P}_{nrS} \mathbf{a}]^{-1} \quad (4.30)$$

where $\mathbf{a} \in \Re^{(m+n+1) \times n}$ is the gradient of the output with respect to the parameters ($\mathbf{w}_{nrS}$), $\mathbf{R} = r_0 I_{n \times n}$ is the variance of measurement noise and $\mathbf{P}_{nrS} \in \Re^{(m+n+1) \times (m+n+1)}$ is the error covariance matrix of the parameters corresponding to the nearest rule.

The gradient ($\mathbf{a}$) is given by

$$\mathbf{a} = \begin{bmatrix} \frac{\partial \hat{\mathbf{y}}^t}{\partial \boldsymbol{\alpha}_{nrS}} \\ \frac{\partial \hat{\mathbf{y}}^t}{\partial \boldsymbol{\mu}_{nrS}^l} \\ \frac{\partial \hat{\mathbf{y}}^t}{\partial \sigma_{nrS}^l} \end{bmatrix} \quad (4.31)$$

and

$$\frac{\partial \hat{y}_j^t}{\partial \alpha_{j,nrS}} = \bar{\phi}_{nrS}, \quad j = 1, 2, \dots, n \quad (4.32)$$

$$\frac{\partial \hat{y}_j^t}{\partial \mu_{nrS,i}^l} = 2\phi_{nrS} \frac{x_i^t - \mu_{nrS,i}^l}{\sigma_{nrS}^2} \frac{\alpha_{j,nrS} - \hat{y}_j^t}{\sum_{k=1}^K \phi_k}, \quad j = 1, 2, \dots, m \quad (4.33)$$

$$\frac{\partial \hat{y}_j^t}{\partial \sigma_{nrS}^l} = 2\phi_{nrS} \frac{\|\mathbf{x}^t - \boldsymbol{\mu}_{nrS}\|^2}{\sigma_{nrS}^3} \frac{\alpha_{j,nrS} - \hat{y}_j^t}{\sum_{k=1}^K \phi_k} \quad (4.34)$$

The error covariance matrix ($\mathbf{P}_{nrS}$) is updated as

$$\mathbf{P}_{nrS} = [I_{z \times z} - \mathbf{G}\mathbf{a}^{tT}] \mathbf{P}_{nrS} + q_0 I_{z \times z} \quad (4.35)$$

where, $z = m + n + 1$ and the scalar quantity (q_0) determines the allowed step in the direction of gradient vector. Note that, I is an identity matrix.

Rule Pruning Criterion: If the contribution of a rule (β_k) in the same class lesser is than a pruning threshold E_p for N_w consequent samples in the same class then that rule is insignificant and can be removed from the network. The contribution of the rule (β_k) is measured as

$$\beta_k = \frac{E^t}{\psi_c} \phi_k \max |\boldsymbol{\alpha}_k|, \quad (4.36)$$

where E^t is the maximum hinge error, ψ_c is spherical potential of the actual class (c), ϕ_k is the firing strength of the k th rule and $\boldsymbol{\alpha}_k$ is the output weight of the k th rule.

When a new rule is added to McFIS-classifier, the error covariance matrix is updated as

$$\begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & p_0 I_{z \times z} \end{bmatrix} \quad (4.37)$$

where p_0 is the initial estimated uncertainty. On the other hand, when a rule is pruned from the network, the dimensionality of error covariance matrix is reduced by removing the respective rows and columns of the matrix $\mathbf{P}$, i.e., remove $(k - 1)(m + n + 1) + 1$ to $k(m + n + 1)$ rows/columns of the matrix $\mathbf{P}$.

- *Sample reserve strategy:* If the current sample does not satisfy either the rule growing condition or the parameter update condition, then the sample is pushed to the rear-end of the training data stream. Since McFIS-classifier employs self-adaptive thresholds, these samples may be used at a later stage in the learning process.

The ‘sample deletion strategy’ helps in achieving the meta-cognitive task of *what-to-learn* by deleting those samples that does not contain any new information. ‘Sample learning strategy’ and ‘sample reserve strategy’ fulfil the *how-to-learn* and *when-to-learn*

aspects of the meta-cognition by either learning immediately or deferring the learning process to a later stage. Learning is carried out either by updating the parameters of the nearest rule or by adding a new rule. Also, McFIS-classifier ensures that the rule base is compact by deleting those rule with insignificant contributions.

For a better understanding, the essence of the meta-cognitive learning aspects are summarized as pseudo-code in Algorithm 2.

```

while all samples are learnt do
  1. Calculate the absolute maximum error  $E^t$ , significance  $\psi$  and
    predicted posterior probability  $\hat{p}(i|\mathbf{x}^t)$  for the current input  $(\mathbf{x}^t, \mathbf{y}^t)$ .
  2. if  $c^t == \hat{c}^t$  AND  $\hat{p}(\hat{c}|\mathbf{x}^t) \geq E_d$  then
    delete the sample without learning. Increment  $t=t+1$ . Goto step 1.
  3. else if  $E^t > E_a$  AND  $c^t \neq \hat{c}^t$  AND  $\psi_c < E_s$  then
    Add new node. Update self-adaptive threshold and EKF parameters
    according to Eqs. ((4.21), (4.37)).
  4. else if  $E^t \geq \hat{E}$  AND  $c^t == \hat{c}$  then
    Update the parameters of the network according to Eqs. (27-33).
    Update the self-adaptive threshold according to Eqn. (26).
  5. else
    Reserve the sample to be learnt later. Increment  $t=t+1$ . Goto step
    1.
  end
  6. if Contribution of a rule according to Eqn. (34) for the same class is
    less than  $E_p$  for  $N_w$  consequent samples then
    Rule is pruned from the network.
  end
  Increment  $t=t+1$ . Goto step 1.
end

```

Algorithm 2: Pseudo-Code for McFIS Classifier

4.2.4 Discussion on the Influence of Self-Regulatory Thresholds

In this section, we explain the influence of the fixed and self-regulatory threshold on the decision making ability of McFIS-classifier and provide a guideline for their initialization. There are four main thresholds which are introduced in this work: delete threshold (E_d), class-specific novelty threshold (E_s), self-regulatory adding threshold (E_a) and self-regulatory parameter update threshold (E_l). One should note that the proposed algorithm employs hinge-loss function as in Eq. (4.11). As a result, the maximum

absolute error (E) can be between 0-2. In here, if the error is greater than 1, then the sample has been misclassified. Otherwise, the sample has been correctly classified, with a lower confidence of classification. This information is used to initialize the add, learn and delete thresholds.

- Delete threshold (E_d): If estimated posterior probability is greater than E_d and predicted class label is same as actual class label, then the current sample is similar to existing knowledge in McFIS-classifier. These samples can be deleted without learning. Posterior probability in Eq. (4.13) is in the range $[0\ 1]$. If E_d is chosen close to 0, most of the samples will be deleted without learning resulting in under-training. Similarly, if E_d is chosen close to 1, most of the samples will be used in learning, resulting in over-training. In general, E_d is chosen in the range $[0.9\ 0.95]$.
- Class-specific novelty threshold (E_S): This threshold measures novelty present in the sample. The range of spherical potential can be between 0-1. If spherical potential is close to 1, then knowledge in the current sample is similar to the existing knowledge in the fuzzy system, otherwise it is novel. If one keeps the threshold (E_S) close 0, then the algorithm will not allow addition of rule. Similarly, if one keeps the threshold (E_S) close to 1, then all sample will be identified as novel sample. Hence, the class specific novelty threshold is recommended in the range $[0.4\ 0.7]$.
- Self-regulatory adding threshold (E_a): The main idea behind using self-adaptive threshold is to identify and learn the samples which have significant knowledge first and then fine tune the parameters of McFIS-classifier using the samples with less significant information. In McFIS-classifier, since we use hinge-loss function as error function Eq. (4.11), the minimum possible prediction in case of misclassification is 1. Therefore self-regulatory threshold, E_a should be greater than 1. If E_a is selected close to one, all misclassified samples will be used for rule addition. If one selects the threshold close to 2, then very few rules will be added and the resultant network may not approximate the decision surface, accurately. Hence, the suggested range for this threshold is $[1.2\ 1.7]$. Note that the meta-cognitive component adapts this threshold such that sample with higher error are learnt first, followed by sample with lower error.

The rate of self-regulation is controlled by the slope parameter δ . The slope parameter controls the rate at which the self-regulatory thresholds reach their minimum. In general, slope is initialized close to 1. Larger the slope, slower the self-adaptive parameters reach their minimum and fewer rules get added to the network.

- Self-regulatory parameter update threshold (E_l): The network should learn a sample when the predicted class and actual class are the same and absolute error is greater than E_l . If one selects the threshold E_l close to 1, then no sample will be used in learning and hence the resultant network do not approximate the decision function accurately. If E_l is selected close to 0, then all samples will be learnt, leading to over-fitting. The samples which produce higher error are first used to tune the parameters and hence the threshold is initialized in the range [0.3 0.6].

One can choose these parameters within suggested range or can perform grid search in this range using cross-validation with training samples.

4.3 Performance Evaluation of McFIS-classifier

In this section, we first present the performance comparison of the proposed McFIS-classifier with different existing sequential learning neuro-fuzzy inference systems including evolving classifier [17] (eClass), sequential adaptive fuzzy inference system [169] (SAFIS), online sequential fuzzy extreme learning machine [168] (OS-Fuzzy-ELM), batch learning extreme learning machine [84] (ELM) and support vector machines [39] (SVM) with radial basis function kernels, using real-world benchmark binary/multi-category classification data sets from the UCI machine learning repository [26].

Next, we highlight specifically the benefits of meta-cognition by using existing SAFIS algorithm on all benchmark data sets.

4.3.1 Dataset Description

For our performance evaluation, we have chosen both small/large samples, low/high dimensional features, and balanced/unbalanced data sets in both binary/multi-category classification problems. The detailed descriptions of 5 binary and 9 multi-category classification data sets are given in Table. 4.1. Note that the data sets are taken from UCI

Table 4.1: Dataset Specifications for Classification using McFIS-Classifier

Data set	No. of features	No. of classes	No. of Samples		I.F.		Random trial
			Training	Testing	Training	Testing	
Liver	6	2	200	145	0.17	0.14	Y
PIMA	8	2	400	368	0.22	0.39	Y
BC	9	2	300	383	0.26	0.33	Y
ION	34	2	100	251	0.28	0.28	Y
Leukemia	7129	2	38	34	0.42	0.18	N
Iris	4	3	45	105	0	0	Y
IS	19	7	210	2100	0	0	Y
Wine	13	3	60	118	0	0.29	Y
GCM	98	14	144	46	0	0.39	Y
Sat	6	9	9108	262144	0	0.88	N
Letter	16	26	13333	6667	0.06	0.09	N
AE	5	4	62	137	0.1	0.32	Y
VC	18	4	424	422	0.1	0.12	Y
GI	9	6	336	105	0.72	0.77	Y

machine learning repository, except for satellite imaging [209], global cancer map (GCM) [209] and acoustic emission [149]. The sample imbalance in training/ testing is measured using imbalance factor (I.F.) and is defined as

$$I.F. = 1 - \frac{n}{N_T} \min_{j=1,2,\dots,n} N_j \quad (4.38)$$

where, N_j is number of samples in class j and $N_T = \sum_{j=1}^n N_j$. If I.F. is zero then training samples contain equal distribution of samples from each class. If I.F. close to one implies a class have fewer samples then the rest.

For efficient comparison, we present them under the following categories as described below:

- Binary Problems: All data sets considered for binary problems have high sample imbalance and are grouped into two categories:
 - Low dimension: Liver, PIMA Indian diabetes (PIMA) and breast cancer (BC) are having smaller number of features with relatively smaller training samples
 - High dimension: Leukemia and Ionosphere (ION) data sets are having smaller samples with high dimensional features.

- Multi-category Problems: The 9 data sets are grouped into three categories:
 - Balanced: Iris, image segmentation (IS) and wine recognition problems have equal number of samples per class in the training set. These data sets are having varying number of features and samples.
 - Unbalanced: Acoustic emission (AE), vehicle classification (VC) and glass identification (GI) data sets are having smaller number of features and samples with a higher class imbalance. The global cancer map (GCM) is having large number of features with higher sample imbalance.
 - Large sample: Satellite imaging [26] and letter recognition are having relatively large number of samples and class labels.

In our study, 10 random trials are generated by maintaining the imbalance factor for all data sets in MATLAB R2011b environment in a quad core system with 16GB RAM in Linux environment.

The tunable parameters of SAFIS, McFIS-classifier, eclass, SVM, ELM, and the number of rules in OS-Fuzzy-ELM are chosen using a 10 fold cross-validation using training samples. The cost parameter and kernel parameter of SVM are optimized using a grid search. The results reported here are for the best combination of parameters for which, the algorithm produces higher validation accuracy.

4.3.2 Performance Measures

The performances of each algorithm on 10 different random trials are measured using both overall accuracy (η_o) and geometric mean accuracy (η_g) for multi-category classification problems. They are defined as

$$\eta_o = 100 \times \frac{1}{N_T} \sum_{j=1}^n q_{jj} \quad (4.39)$$

$$\eta_g = \sqrt[n]{\prod_{j=1}^n \left(100 \times \frac{q_{jj}}{N_j} \right)} \quad (4.40)$$

where q_{jj} is the number of samples correctly classified in class j .

In addition to the above mentioned measures, we have also used the F -measure for binary classification problems as defined in [89]. Since, the proposed McFIS-classifier selects samples for training, we have also included the percentage samples used in training (PS) as an additional measure. This is defined by

$$PS = \frac{\text{number of samples used for training}}{\text{total number of samples available for training}} * 100\%$$

4.3.3 Performance Comparison

The mean and standard deviation of the number of rules, overall accuracy, geometric mean (F-measure for binary problems) accuracy for different algorithms on the benchmark data sets are given in Table 4.2 and 4.3. It can be seen that the proposed McFIS-classifier achieves better performance in all problems with smaller number of rules and samples.

From the performance comparison result for binary classification problems given in Table 4.2, one can see that the proposed McFIS-classifier is approximately 2% higher than the next best performing eclass on low dimensional data sets with smaller number of rules. In case of higher dimensional problems like ION and Leukemia, McFIS-classifier outperforms the rest of the classifiers by 3-6%. In case of simple BC problem, all classifiers achieve almost same performance as that of McFIS-classifier. But McFIS-classifier requires only 16% of total samples to develop the classifier and also employs lesser number of rules (4 rules) to approximate the decision surface. This clearly indicates the advantage of proposed meta-cognitive learning principle.

In case of balanced data sets (IS, Wine and IRIS), McFIS-classifier uses only 45-60% training to achieve 2-3% improvement over other algorithms with the lowest number of rules. Meta-cognitive sample deletion criteria helps in removing redundant samples from the training set and thereby improving the generalization performance. For highly unbalanced data sets, one can see that the proposed McFIS-classifier is able to achieve significantly better performance than the rest of the classifier. Results for SVM and ELM for large data sets are taken from [209]. For large data sets, McFIS-classifier achieves similar performance as that of SVM and ELM classifiers with minimal number of rules.

Table 4.2: Performance Evaluation of McFIS-classifier with Other State-of-the-art Methods on Binary Classification Problems

Data set	Algo rithm	CPU (sec)	Rules		Testing					
					η_o		η_g		F-measure	
			Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
Liver	SVM	0.3	157.5 ^a	4.72	69.21	2.10	62.91	4.17	0.75	0.02
	eClass	0.12	79.4	3.15	69.50	3.10	69.79	3.39	0.75	0.03
	ELM	0.12	127.0	15.67	64.55	3.80	63.52	4.04	0.69	0.04
	OS-FUZZY-ELM	10.5	79.9	3.24	64.27	3.40	62.50	4.05	0.69	0.04
	SAFIS	2.17	49.0	4.04	60.68	6.63	53.69	11.23	0.64	0.17
	McFIS-classifier ^b	0.56	79.4	6.77	72.89	1.95	70.49	2.19	0.77	0.02
PIMA	SVM	0.19	252.7	42.28	76.76	1.45	66.78	6.58	0.57	0.07
	eClass	0.26	419.0	20.39	75.41	3.07	70.52	4.62	0.58	0.04
	ELM	0.2	172.0	25.73	70.86	1.44	65.50	2.67	0.54	0.03
	OS-FUZZY-ELM	40	65.3	5.90	69.40	1.58	62.34	3.10	0.49	0.04
	SAFIS	0.92	48.1	4.34	69.86	3.70	61.30	7.80	0.50	0.09
	McFIS-classifier ^c	0.25	109.9	10.95	76.63	0.78	72.66	1.80	0.63	0.02
BC	SVM	0.04	27.7	3.60	96.55	0.57	96.41	0.67	0.94	0.01
	eClass	0.18	217	5.19	96.73	0.59	96.51	0.72	0.95	0.01
	ELM	0.12	37.9	1.34	96.97	0.47	96.89	0.87	0.95	0.01
	OS-FUZZY-ELM	44	14.9	2.76	87.23	0.64	81.17	1.67	0.78	0.02
	SAFIS	0.9	10.2	3.22	96.34	1.08	96.63	1.27	0.95	0.02
	McFIS-classifier ^d	0.27	3.90	0.32	97.73	0.41	98.09	0.36	0.97	0.01
ION	SVM	0.20	70.9	10.27	91.24	1.11	90.10	2.69	0.87	.024
	eClass	0.18	26.3	3.80	63.78	10.82	67.18	10.62	0.42	0.19
	ELM	0.13	46	1.76	80.26	2.30	74.63	2.59	0.69	0.03
	OS-FUZZY-ELM	140.0	35.2	10.10 ^p	78.80	3.66	72.06	5.26	0.66	0.07
	SAFIS	0.79	27.9	9.09	78.80	5.30	71.76	12.0	0.65	0.14
	McFIS-classifier ^e	0.26	19.0	2.95	93.27	1.00	93.02	1.30	0.90	0.02
Leukemia	SVM	-	-	-	82.34	-	-	-	-	-
	LSSVM	-	-	-	85.29	-	-	-	-	-
	ELM	-	-	-	82.35	-	-	-	-	-
	McFIS-classifier ^f	-	12	-	91.17	-	90.37	-	-	-

(a) # support vectors (b) P.S: $71.75 \pm 7.5\%$ (c) P.S: $63.8 \pm 4.3\%$ (d) P.S: $15.8 \pm 6.7\%$ (e) P.S: $53.8 \pm 15.8\%$ (f) P.S: 97.8%

Table 4.3: Performance Evaluation of McFIS-classifier with Other State-of-the-art Methods on Multi-class classification problems

Data set	Algorithm	CPU (sec)	Rules		Testing			
					η_o		η_g	
			Mean	S.D.	Mean	S.D.	Mean	S.D.
IS	SVM	0.1	107.3	10.91	90.92	0.43	90.13	0.46
	eClass	0.28	49.7	6.79	87.74	1.98	87.21	2.25
	ELM	0.22	55.5	10.91	90.13	0.72	89.4	0.83
	OS-FUZZY-ELM	43.1	82.5	18.90	84.07	1.22	83.42	1.17
	SAFIS	2.1	82.3	18.90	85.65	2.90	83.76	3.87
	McFIS-classifier ^a	0.44	55.3	7.40	91.57	0.32	90.63	0.79
Wine	SVM	0.1	37.0	3.02	96.02	1.20	96.77	0.93
	eClass	0.3	60.0	0	93.10	1.99	92.58	2.03
	ELM	0.25	15.2	2.09	94.83	1.35	95.58	1.08
	OS-FUZZY-ELM	44.0	14.5	6.15	92.30	1.62	92.20	1.49
	SAFIS	1.8	9.3	4.5	93.47	2.85	94.21	2.72
	McFIS-classifier ^b	0.4	6.0	0	99.83	0.53	97.99	0.0
Iris	SVM	0.02	16.6	0.93	95.99	1.17	95.90	1.22
	eClass	0.03	7.0	0.82	96	1.95	96.0	1.83
	ELM	0.01	16.0	4.59	96.38	1.08	96.29	1.16
	OS-FUZZY-ELM	0.03	14.3	0.63	88.57	7.78	88.06	8.52
	SAFIS	0.32	10.0	3.59	95.42	1.54	95.29	1.62
	McFIS-classifier ^c	0.14	6.66	1.90	97.56	0.96	97.52	0.99
AE	SVM	0.11	21.7	4.48	97.66	1.02	97.62	1.43
	eClass	0.1	61.8	0.63	97.52	0.85	98.30	0.50
	ELM	0.16	14.5	3.68	97.88	0.87	97.34	1.11
	OS-FUZZY-ELM	0.17	11.7	2.54	93.20	3.86	91.72	4.80
	SAFIS	0.32	10	3.59	95.42	1.54	95.29	1.62
	McFIS-classifier ^d	0.25	7	0.94	98.80	0.38	98.36	0.61
GCM	SVM	0.02	120.6	5.78	76.53	2.80	63.91	5.59
	eClass	0.4	114.4	10.20	76.39	3.89	58.16	2.97
	ELM	0.22	56.5	4.11	91.80	2.83	63.39	4.71
	OS-FUZZY-ELM	180	56.0	4.18	62.03	3.97	55.29	4.96
	SAFIS	1.58	50.6	4.90	68.88	5.28	64.60	4.97
	McFIS-classifier ^e	1.64	50.2	3.42	88.56	2.32	74.10	6.24
VC	SVM	1.8	246.8	58.20	73.20	2.01	69.4	3.61
	eClass	0.56	153.0	21.82	65.67	2.82	65.03	2.26
	ELM	0.36	150.0	25.30	77.09	3.44	75.16	3.83
	OS-FUZZY-ELM	97.2	132.80	11.54	76.83	2.59	69.05	4.45
	SAFIS	7.4	288.2	19.20	77.03	1.11	66.96	1.25
	McFIS-classifier ^f	4.4	186	7.20	75.80	2.25	73.44	2.12
GI	SVM	0.92	176.6	18.90	77.23	5.01	85.83	3.62
	eClass	0.35	183.0	5.52	85.90	2.53	90.48	4.23
	ELM	0.3	86.0	9.17	80.09	3.05	88.81	2.28
	OS-FUZZY-ELM	13.2	70.0	8.49	85.57	3.01	90.01	1.86
	SAFIS	3.1	57.3	2.79	61.04	8.33	70.64	7.69
	McFIS-classifier ^g	0.94	64.3	6.36	87.62	2.68	92.88	1.71
Letter	SVM	-	-	-	92.87	-	-	-
	ELM	-	-	-	93.51	-	-	-
	McFIS-classifier ^h	-	2015	-	93.21	-	92.11	-
Sat	SVM	-	1298	-	92.21	-	-	-
	OS-ELM	-	1500	-	75.32	-	-	-
	McFIS-classifier ⁱ	-	1159	-	90.15	-	91.04	-

(a) P.S: $61.2 \pm 4.9\%$ (b) P.S: $60 \pm 4\%$ (c) P.S: $46 \pm 11\%$ (d) P.S: $27.4 \pm 7.4\%$ (e) P.S: $96.5 \pm 3.2\%$
(f) P.S: $93 \pm 1.3\%$ (g) P.S: $67.35 \pm 9.2\%$ (h) P.S: 98.2% (i) P.S: 98.2%

Overall, one can say that meta-cognitive learning with class specific criterion and hinge loss error helps McFIS-classifier to approximate the decision function more efficiently.

4.3.4 Statistical Comparison

In order to compare the performance of the proposed McFIS-classifier over other algorithms on different benchmark data sets, we employ an one-way repeated ANOVA measure followed by a pair wise comparison using post-hoc Dunnett test [141]. The ANOVA measure compares whether the mean of individual experimental condition differs significantly from the aggregate mean across all conditions. If the measure F score is greater than the F -statistics at 95% confidence level then one rejects the equality of mean hypothesis (the algorithms used in our study perform similarly on different data set).

If one-way repeated ANOVA test reject the equality hypothesis then pair-wise post-hoc should be conducted to test which mean is different from others. In our study, we have used 6 different algorithms and 11 different data sets. For a given data set and an algorithm, 10 random trials are conducted to measure the mean and variance of the algorithms performance. The performances of six classifiers on ten data sets are organized as six groups and ANOVA monitors three kinds of variations in the data, viz., within-group variations, between-group variations and the total variation. The F -score obtained using repeated measures of one-way ANOVA test is 6.4491, which is greater than the F -statistic at 95% confidence level ($F_{5,45,0.05}$ is 1.68), i.e, $6.4491 > 1.68$. Hence, one can reject equality of mean hypothesis at a confidence of level 95%.

Next, we conduct a pair-wise comparison using a parametric Dunnett test to highlight the performance significance of McFIS-classifier with respect to other classifiers. Here, the proposed McFIS-classifier is used as a control. The t -observed obtained for individual pairs are: McFIS-SVM: 1.882, McFIS-e-Class: 2.922, McFIS-ELM: 2.283, McFIS-OS-Fuzzy-ELM: 4.53, McFIS-SAFIS: 4.553. Note that the t -observed for all pairs are greater than t -critical (t -critical is 1.68 at 95% confidence level). Hence, we can say that McFIS-classifier performs better than other classifiers used in this study with a confidence of 95%.

4.4 Discussion on the Effect of Meta-Cognition

In this section, we analyze the effect of meta-cognitive components of the learning algorithm and Gaussian rule exploitation using SAFIS algorithm as a baseline. For this purpose, we have conducted three different experiments on SAFIS using 10 different data sets.

- Effect of Gaussian rule exploitation and hinge error: The parameter initialization for a new rule (μ_{K+1} , σ_{K+1} and α_{K+1}) in SAFIS is replaced using the proposed approach given in Eqs. (4.24)-(4.25). The error function is modified to hinge loss error as defined in Eq. (4.11). We call this algorithm as SAFIS*.
- Effect of sample selection: The SAFIS* algorithm is trained using samples selected by McFIS-classifier, i.e., the selected samples are sequentially presented in the same order in which they are used in McFIS-classifier. We call this algorithm as SAFIS[†].
- Effect of non-deletion of samples: The SAFIS* algorithm is trained using samples selected by McFIS-classifier followed by deleted samples. We call this algorithm as SAFIS[‡].

The first experiment emphasizes the advantage of Gaussian rule exploitation along with the use of hinge loss error function. The next two experiments emphasize the importance of sample selection/deletion in the learning process. The training/testing overall and geometric mean accuracies and number of rules are given in Table 4.4 for all the 10 benchmark data sets. From the results, one can make the following observations

- SAFIS* achieves 2-6% improvement in testing over SAFIS classifier. This clearly shows that hinge loss error and Gaussian exploitation improves the performance significantly.
- SAFIS[†] achieves 2-10% improvement over SAFIS* with a smaller rules. On low dimensional and unbalanced cases, the performance is similar to that of SAFIS*. Since, the rule deletion criterion in SAFIS is not class specific, it affects the performance in unbalanced data sets significantly.

Table 4.4: Quantitative Verification of McFIS Learning Philosophy in SAFIS Algorithm [169] using Classification Problems

Data set	Algorithm	Rules	Training η_o	Training η_g	testing η_o	Testing η_g^i
Liver	SAFIS	129	83	81.44	64.82	63.97
	SAFIS*	94	83.5	82.21	66.2	64.15
	SAFIS [†]	78	83.5	81.44	65.51	65.39
	SAFIS [‡]	82	83.5	82.21	66.2	64.15
PIMA	SAFIS	58	67	65.59	68.2	66.95
	SAFIS*	55	73.5	68.28	69.29	64.96
	SAFIS [†]	53	67.25	71.48	77.17	69.23
	SAFIS [‡]	57	76	71.28	70.65	70.44
BC	SAFIS	6	95	95.47	95.82	96.66
	SAFIS*	5	97	96.5	97.39	97.65
	SAFIS [†]	5	96	95.71	97.91	98.04
	SAFIS [‡]	6	96.5	97.39	95.82	96.67
ION	SAFIS	35	66	61.28	72.9	69.81
	SAFIS*	32	83	86.72	70	75.97
	SAFIS [†]	29	89	91.4	80.87	84.6
	SAFIS [‡]	32	91	91.4	78.88	75.95
IS	SAFIS	82	95.23	95.23	82.14	82.14
	SAFIS*	60	95.24	95.24	88.47	88.47
	SAFIS [†]	45	96.67	96.67	90.86	90.86
	SAFIS [‡]	48	96.67	96.67	88.47	88.47
Wine	SAFIS	13	100	100	95.76	96.73
	SAFIS*	12	100	100	96.61	97.38
	SAFIS [†]	11	100	100	98.3	97.6
	SAFIS [‡]	11	100	100	96.61	95.24
Iris	SAFIS	12	100	100	94.28	94.28
	SAFIS*	10	100	100	96.16	96.16
	SAFIS [†]	9	100	100	96.19	96.19
	SAFIS [‡]	10	100	100	96.19	96.19
AE	SAFIS	6	95.55	95.55	95.23	95.23
	SAFIS*	6	98.38	98.43	99.27	99.17
	SAFIS [†]	5	98.38	98.43	99.27	99.17
	SAFIS [‡]	6	98.38	98.43	99.27	99.17
VC	SAFIS	309	77.12	77.85	68.25	69.28
	SAFIS*	272	77.45	76.39	72.17	70.68
	SAFIS [†]	251	76.88	76.73	69.66	70.32
	SAFIS [‡]	270	77.12	77.85	68.24	69.28
GI	SAFIS	87	73.3	86.75	66.66	82.55
	SAFIS*	83	84.4	86.75	66.67	84.16
	SAFIS [†]	73	76.15	87.15	71.42	87.61
	SAFIS [‡]	80	84.4	86.75	71.43	84.16

- Since SAFIS[†] uses all samples (including the deleted samples), its training performance is significantly better than rest of the algorithms. But, the testing efficiency is slightly lower than that of SAFIS[†].

The above results clearly highlight that the use of meta-cognitive learning, Gaussian rule exploitation and hinge loss error improves the performance of SAFIS significantly.

4.5 Summary

In this chapter, a Meta-Cognitive sequential learning algorithm for a Neuro-Fuzzy Inference System was proposed to solve classification problems, efficiently (McFIS-classifier). The meta-cognitive component in McFIS helps the system choose a suitable learning strategy for neuro-fuzzy classifier. Also, the learning strategies consider a sample overlapping condition and novelty of the current sample for proper initialization of a new rule, that helps in minimization of misclassification rates. The meta-cognitive learning adapts the learning strategies appropriately and hence it decides *what-to-learn*, *when-to-learn* and *how-to-learn*, efficiently. The performance of the proposed McFIS-classifier has been evaluated using a number of benchmark binary as well as multi-category classification problems, with a wide range of class imbalance. The qualitative and quantitative performance analysis using multiple data sets clearly indicates the superior performance of the proposed McFIS-classifier.

Next, we evaluate the performance of the proposed McFIS on two real world video analytics problems of human action recognition and facial emotion recognition. In the next chapter, we present a McFIS based human action recognition system. It employs optical flow based features to represent human action.

Chapter 5

Meta-Cognitive Neuro-Fuzzy Inference System based Classifier for Human Action Recognition

In the previous chapter, a Meta-Cognitive sequential learning algorithm for a Neuro-Fuzzy Inference System based classifier (McFIS-classifier) was proposed for efficiently solving classification problems. It employed meta-cognitive principles of self-regulation along with class-specific knowledge for finding accurate decision boundaries based on training data. In this chapter, McFIS-classifier is employed for solving an important video analytics problem of human action recognition. Two kinds of action recognition problems are considered: action recognition based on 2D video and action recognition based on 3D video sequences. The aim of McFIS-classifier in both these cases is to find an optimal decision boundary separating various actions based on the given input video sequence.

5.1 Overview

In this chapter, a McFIS-classifier based action recognition system is proposed. The Fig. 6.1 gives an overview of the proposed action recognition system. Action recognition based on McFIS-classifier consists of two parts: training and testing. During the training process, first the raw video is pre-processed. During pre-processing, the the objects of interests are identified and video is segmented to contain a single type of action. This is followed by action feature extraction. During this step, features which are representative

Figure 5.1: Overview of McFIS based Action Recognition System

of the actions are extracted. The next step involves training. McFIS-classifier learns the functional relationship between the annotated action features and the actions they represent. The rules representing the functional relationships are stored in the rule base.

During testing, similar to the training phase, raw video is pre-processed, objects of interest identified, segmented, and action features are extracted. McFIS-classifier employs the rules in the rule base to predict the actions based on the extracted action features. Various feature extraction techniques are available in the literature of computer vision. A review about the same is provided in Sections 5.2.2 and 5.3.2. The extracted features are processed by McFIS-classifier to predict the corresponding actions.

In the next section, we shall describe the 2D action recognition process.

5.2 Human Action Recognition based on 2D Video Sequence

5.2.1 Motivation

Recognizing human action is an important aspect in various machine vision applications such as, video analytics, gesture recognition, video indexing, and man-machine interface. It is, in its own respect, one of the most promising, yet immensely difficult task. The challenge in development of human action recognition system is two fold: (a) extraction of information rich features (action representation) and (b) efficient mapping of features to each action (action recognition). The features and the algorithm chosen for the recognition of human action should be able to overcome the problems arising due to self-occlusion, cluttering, viewpoint change, intra-class variations, etc. Various approaches are available which try to tackle the above problems [88, 237, 159]. However, few works employ a combination of local and global feature descriptors. It has been shown in literature that both local as well as global level information is required for effective modeling and representation of action. A review of major action representation and recognition research is presented in Section 5.2.2. It has been reported that optical flow vectors [49] can represent apparent movement in object, naturally. Moreover, it can represent simple and dynamic features hierarchically from local pixel level to global object level. Therefore, in this chapter, we represent motion features corresponding to each action by extracting the accumulated motion information (optical-flow) over a small time window.

The aim of any action recognition system is to find the functional relationship between the feature vectors and the action classes. Machine learning algorithms are bestowed with the ability to learn the functional relationship between the inputs and the output, from examples [100, 12]. Since the video for action recognition almost always arrive sequentially and are massive in size, we require algorithms which can learn from stream of data in an online fashion. Artificial neural networks [179, 211, 177] with their excellent learning algorithms such as [175, 2] and fuzzy inference systems [88, 116] with their knowledge representation ability are two commonly employed machine learning techniques for such problems.

In recent times, another type of networks known as neuro-fuzzy inference systems [120, 241, 169, 234, 219, 108] which combines the predictability and interpretability characteristics of neural networks and fuzzy inference systems are gaining popularity in research community. Furthermore, these neuro-fuzzy inference systems have been employed for solving problems in various domains such as load forecasting [71], system modeling [253, 66], industrial control [157, 102], etc. A few works can also be found wherein neuro-fuzzy systems are employed for human activity prediction [88]. Works are also available in literature which combine properties of wavelets into the learning of fuzzy networks[99, 3, 95]. A review on existing neuro-fuzzy inference systems is provided in Chapter 2.

In this chapter, Meta-Cognitive sequential learning algorithm for a Neuro-Fuzzy Inference System based classifier (McFIS-classifier) is employed to efficiently learn the underlying decision boundary separating various actions. The work done in this chapter has been published as an article in [198].

In order to validate the performance of McFIS for learning optical-flow features, we perform a comparative study with support vector machine (SVM) [39] with Gaussian kernel and other state-of-the-art methods. These methods will be validated for Weizmann data set[68] and KTH data set[184]. For Weizmann data set, about 70% of the data will be employed for training and rest for testing, whereas for KTH data set, about 50% of data will be employed for training. Optical flow features are extracted from the given data set for recognition and are employed with McFIS and SVM. The results indicate better classification ability of McFIS classifier. The advantage of optical flow features with proposed McFIS classifier are clearly illustrated by comparing the performance with other approaches reported in literature.

5.2.2 Background and Related Works on 2D Action Recognition

In most of the feature representation techniques, one of the first steps involved is foreground detection [121]. Earlier approaches for action representation involved image based representation, where the region of interest obtained upon background subtraction were employed for feature extraction. Such a technique was employed by Goldenberg et. al [64], where Eigen shapes of foreground silhouettes is extracted for representation. The

drawback of such methods is their lack of information regarding the motion of the silhouettes. Polana and Nelson [158] and Seitz and Dyer [185], tap in techniques based on the periodicity analysis. But these techniques are limited to recognition of cyclic actions.

In recent times, research works have concentrated on analyzing actions in a video sequence by considering them as space-time volumes [250, 112] of gradients, intensities, etc. In addition, some of the recent works in the field of object recognition have demonstrated that silhouettes contain detailed information about the shape of objects[24, 68]. In their initial work, Bobick and Davis [27] extracted silhouettes from video sequence and constructed binary motion energy image which indicates where motion occurs. Motion history images are also constructed, where the pixel intensities are recency function of the silhouette motion. The two templates are compared using *Hu* moments [83]. These space-time volume over silhouette images have been proven to have the capability to handle phase correlation. Gorelick et. al [68] stacked silhouettes over a given sequence to form space-time volume. Then the solution of the Poisson equation is used to derive local space-time saliency and orientation features. Global features for a given temporal range are obtained by calculating weighted moment over these local features. These features are then used to recognize actions using nearest neighbor classifiers. In literature [220], Histogram of Oriented Gradients (HOG) [44] has also been employed to focus on foreground edges by non-negative matrix factorization. However, it suffers from high dimensionality. The high dimensionality of HOG features are reduced using principal component analysis [123].

Authors have also extracted feature vectors from segments of silhouette tunnels [69] where, the classification is performed by approximating the log-covariance of query segment by sparse linear combination of action dictionaries. Efros et al. [49] used blurred optical flows to recognize actions of players in football game occupying a very small region in image plane. The motion descriptors are constructed from the positive and negative components of horizontal and vertical flow vectors.

In contrast to global representation, local representation provide better robustness to viewpoint variation, occlusion and noisy background subtraction. In local representation, the video is encoded as a collection of local patches. Shao et al.[187] represented action using correlogram of body poses obtained from the silhouette sequences. In order to

avoid the issues of viewpoint variations and partial occlusion the region of interest is divided into fixed grids, where each grid describe the image observation locally. Another commonly used local descriptor is 3-dimensional Harris corner detector [112], where the original Harris corner detector is extended to 3-dimension. More details on previous works on action representation and recognition could be availed from the reviews [237, 159].

Most of the widely used techniques rely on computationally expensive features, extracted from space-time volumes or silhouette tunnels or optical flow information, which are not suitable for real-time applications. Optical flow information between consecutive image frames provide vital cue about the object dynamics. In literature, many works could be found which employ either local features such as space-time interest points or global features like silhouettes. But for effective modeling of the action, both local and global level dynamics are required. It has been reported that optical flow vectors can extract simple dynamic features hierarchically from local pixel level to global object level. Therefore, in this chapter, we represent motion features corresponding to each action by extracting the accumulated motion information (optical-flow) over a small time window.

Next section provides the details of optical flow feature extraction from frame level and subsequence level.

5.2.3 Action Representation

One of the most challenging tasks in action recognition is action representation. In many real-life scenarios, background clutter, occlusion/ self-occlusion, scale variations, etc. make action recognition a difficult task. Many works are available which aim to find the best possible representation of action features [20]. The motion information between two frames is a vital source of information to model human actions. In this chapter, motion flow information between consecutive frames are employed for action representation.

In this chapter, motion flow between consecutive frames are obtained by employing Lukas and Kanade's optical flow method [124]. Action is represented here as accumulated motion flow over a brief period of time window (4-6 frames) as given by

$$G_k(i, j) = \sum_n F_n(i, j), \quad n \in \{(k - w), \dots, k\} \quad (5.1)$$

Figure 5.2: Example of Optical Flow Feature Extraction Process with Weizmann Data Set

where, $G_k(i, j)$ is the accumulated flow for the k^{th} frame at $(i, j)^{th}$ location and $F_n(i, j)$ is the optical flow between n^{th} and $(n - 1)^{th}$ frame at $(i, j)^{th}$ location.

For ‘wave2’ action from Weizmann human action data set the features are extracted as in fig. 5.2. The original action is given in fig. 5.2(a). The background subtracted image obtained from the original image is given in fig. 5.2(b). The corresponding flow history image is obtained in fig. 5.2(c). From the background subtracted image, we obtain object center about which the rectangular patches are chosen for feature extraction. This representation is advantageous in recognizing actions instantly from video frames.

Now we shall discuss the feature extraction technique employed in this chapter.

5.2.3.1 Feature Extraction

The optical flow for each frame of video is obtained initially, as shown in section 3.1. The flow vectors of the current frame are accumulated over past few frames (six frames in this study). The object center for each frame is obtained from the mask generated by subtracting the given background video frame from the current frame. In this experiment we employ fifty four rectangular patches of hierarchically increasing sizes, about the object center. The mean value of non-zero motion vectors in each patch is computed as the representative motion vector, given by

$$F_k^b = \frac{\sum_{(i,j) \in R_b} G_k(i,j)}{\sum_{(i,j) \in R_b} I_k(i,j)} \quad (5.2)$$

where,

$$I_k(i,j) = \begin{cases} 1 & \text{if } |G_k(i,j)| > 0, \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

Here, F_k^b indicates the representative motion vector corresponding to b^{th} rectangular block in k^{th} frame, R_b indicates the pixels that belong to b^{th} rectangular block and $I(i,j)$ is a binary matrix indicating the locations of non-zero motion vectors. The final frame-level features are extracted by cascading the representative motion vector of each block as,

$$\mathbf{x}_k^f = [F_k^1 \ F_k^2 \ \cdots \ F_k^{54}] \quad (5.4)$$

The process of frame level feature extraction is depicted in fig. 5.3. The feature vectors are derived by finding the mean optical flow within each of the rectangular patches surrounding the object center, in an hierarchical process. At first, the mean flow vectors are obtained at finer resolution using thirty six windows of size six-by-four, symmetrically placed around the image center (ref. Fig. 5.3(a)). The remaining features are obtained by merging the windows to sizes of 12×8 (Fig. 5.3(b)), 18×12 (Fig. 5.3(c)), 36×12 (Fig. 5.3(d)), 18×24 (Fig. 5.3(e)) and 36×24 (Fig. 5.3(f)).

The final representative motion vectors of all the 54 patches are collected as feature vector. Since each of the motion vector has x and y components, there are totally 108 dimensions.

Figure 5.3: Optical Flow based Feature Extraction Methodology as Employed in this Thesis

For recognizing actions in sub-sequence level, the video is segmented into many overlapping sub shots. In this chapter, each sub shot contains 8 consecutive frames. The feature vector for each sub shot is obtained by combining individual frame features as

$$\mathbf{x}_k^s = [\mathbf{x}_{k-7}^f \cdots \mathbf{x}_k^f] \quad (5.5)$$

where, $\mathbf{x}_k^s$ is the feature vector for k^{th} sub-sequence and $\mathbf{x}_k^f$ is the feature vector of the k^{th} frame. The size of each k^{th} sub-sequence is 108×8 .

In this chapter, all the feature vectors are normalized such that they are in the range $[-1,1]$. The objective of the work is approximate the functional relationship between the local/global optical flow features and action label. In the next section, we shall evaluate the performance of McFIS-classifier in identifying human actions based on optical flow based features extracted from videos.

5.2.4 Performance Evaluation of 2D Action Recognition System

In this section we provide a performance evaluation of the developed McFIS with optical flow features in recognizing human actions efficiently. For performance analysis, two data sets are considered: Weizmann data set [68] and KTH data set [184] for human action recognition. The Weizmann¹ data set contains 90 low resolution video sequences (180×144), where 10 actions, as shown in Fig. 5.4, are performed by 9 subjects. The background videos are also provided for all action sequences. The 10 actions include: bend, jumping-jack (jack), jump-forward-on-two-legs (jump), jump-in-place-on-two-legs (pjump), run, gallop-side-ways (side), skip, walk, wave-one-hand (wave1) and wave-two-hands (wave2). The KTH² data set consists of six actions classes: walking, jogging, boxing, clapping, waving and running, as shown in Fig. 5.5.

For each of the data set, the performance is compared at frame level as well as sub-sequence level. For the performance evaluation of McFIS, the optical flow features extracted from the two data sets are compared with kernel Support Vector Machines (SVM) [39]. For SVM, the parameters C and γ are optimized using grid search technique. Both SVM and proposed McFIS are implemented in Matlab 7.11 on Windows

¹<http://www.wisdom.weizmann.ac.il/vision/SpaceTimeActions.html>

²<http://www.nada.kth.se/cvap/actions/>

Figure 5.4: Illustration of Actions Performed in Weizmann Data Set [68]

Figure 5.5: Illustration of Actions Performed in KTH Data Set [184]

Table 5.1: Performance Comparison of McFIS-classifier with SVM on Frame Level Weizmann Data Set

Action	McFIS		SVM	
	Accuracy Train (%)	Accuracy Test (%)	Accuracy Train (%)	Accuracy Test (%)
Bend	98.8	97.4	24.1	33.1
jack	98.8	96.1	93.3	91.7
Jump	99.6	97.8	63.9	78.1
pJump	99.2	92.3	39.1	48
Run	98.3	96.2	80	87.8
Side	100	95.5	74.2	76.5
Skip	94.8	85.5	76.9	81.2
Walk	100	99.3	90.6	92.8
Wave1	98.9	94.2	76.6	73.8
Wave2	98.7	93.4	40.1	38.4

environment with Pentium duo processor (2.6 GHz) and 4 GB memory. Performance of proposed McFIS classifier is compared with SVM on frame level and subsequence level features. In addition, the significance of optical flow features as feature vector for action recognition problem has been analyzed by considering the performance comparison with various other state-of-the-art methods reported in literature.

5.2.4.1 Recognition at Frame Level

In Weizmann data set, totally 5036 frame level feature vectors are obtained from 10 action classes. For training, 3000 randomly selected frame level features are used and the remaining 2036 frames are used for testing. The performance of the classifier is evaluated using average and overall accuracies. For each action in KTH data set, the sequences without scale change are considered. Totally, 10,398 frame level samples are obtained from 6 action classes. Randomly chosen 5000 samples are used for training and tested using the remaining samples.

The performance for individual action for Weizmann data set and KTH data set is given in table 5.1 and 5.2, respectively. In order to compare the results with regular classifiers, the same data set is used for training multi-class SVM, using lib linear package³. The performance comparison on SVM for Weizmann and KTH data set is given in table

³<http://www.csie.ntu.edu.tw/~cjlin/liblinear>

Table 5.2: Performance Comparison of McFIS-classifier with SVM on Frame Level KTH Data Set

Action	McFIS		SVM
	Accuracy Train (%)	Accuracy Test (%)	Accuracy Test (%)
Boxing	78	79.6	64.2
Hand clapping	92.67	91.8	69.2
Hand waving	81.6	79.3	83.3
Jogging	75.4	62.9	47.9
Running	67.6	42.2	21.9
Walking	92.8	89	52.2

5.1 and 5.2, respectively. From the tables it could be seen that, for Weizmann data set, SVM has an average training and testing efficiency of 65.8% and 69.9%. McFIS outperforms SVM classifier by about 30%. It should be noted that McFIS employs 450 rules and 90% of the total training samples to achieve this accuracy. Similarly for KTH data set, McFIS outperforms SVM by about 20%. In both these scenarios, the meta-cognitive components of McFIS along with the spherical potential has helped learn the information contained in the sample better.

5.2.4.2 Recognition at Sub-sequence Level

For sub-sequence level recognition, in Weizmann data set there are 4385 sub-sequences from 10 action classes. For training, 3000 randomly selected sub-sequences are used and the remaining is used for testing. Similarly for KTH data set there are 10,084 sub-sequences from 6 action classes. For training, 5000 randomly chosen sub-subsequence features are used and the performance of McFIS classifier is tested using the remaining sub-sequences.

The testing confusion matrix for Weizmann and KTH data set on sub-sequence level recognition is provided in table 5.3 and 5.4, respectively. In Weizmann data set the misclassification is between jumping-jack and jump-in-place-on-two-legs, which is visually similar. Similarly, skipping, jumping and galloping belong to the same group of actions and Waving of hands belong to a single group.

In case of KTH data set significant misclassification occur among jogging, running and walking classes, which are visually similar except for the speed factor.

Table 5.3: Testing Confusion Matrix of Weizmann Sub-Sequence Level Features Dataset with McFIS-classifier

Action	Bend	Jack	Jump	pJump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	100	0	0	0	0	0	0	0	0	0
Jack	0	99	0	1	0	0	0	0	0	0
Jump	0	0	100	0	0	0	0	0	0	0
pJump	0	1	0	99	0	0	0	0	0	0
Run	0	0	0	0	100	0	0	0	0	0
Side	0	0	0	0	0	100	0	0	0	0
Skip	0	0	1	0	1	1	97	0	0	0
Walk	0	0	0	0	0	0	0	100	0	0
Wave1	0	0	0	0	0	0	0	0	100	0
Wave2	0	0	0	0	0	0	0	0	1	99

Table 5.4: Testing Confusion Matrix of KTH Sub-Sequence Level Features Dataset with McFIS-classifier

Action	Boxing	Clapping	Waving	Jogging	Running	Walking
Boxing	97	3	0	0	0	0
Clapping	1	98	1	0	0	0
Waving	1	3	96	0	0	0
Jogging	0	0	0	89	5	6
Running	0	0	0	20	78	2
Walking	0	0	0	3	0	97

Table 5.5: Performance Comparison of McFIS-classifier with SVM on Sub-Sequence Level Weizmann Data Set

Action	McFIS		SVM	
	Accuracy Train (%)	Accuracy Test (%)	Accuracy Train (%)	Accuracy Test (%)
Bend	99.8	100	100	83.1
jack	99.4	99.3	100	88.2
Jump	100	100	100	78.9
pJump	100	99.4	100	72.8
Run	100	100	100	94.7
Side	99.6	100	100	83.7
Skip	99.2	96.9	99.6	87.3
Walk	100	100	100	95.1
Wave1	99.7	100	100	96.6
Wave2	99.7	99	100	93.0

Table 5.6: Performance Comparison of McFIS-classifier with SVM on Sub-Sequence Level KTH Data Set

Action	McFIS	SVM
	Accuracy Test (%)	Accuracy Test (%)
Boxing	96.6	69.8
Hand clapping	97.6	67.5
Hand waving	96.2	86.8
Jogging	89.4	60.7
Running	77.1	31.9
Walking	96.7	66.2

The performance for individual action at sub-sequence level for Weizmann and KTH data set on state-of-the-art SVM classifier is given in tables 5.5 and 5.6, respectively. For Weizmann data set, it could be seen that the testing performance of McFIS is significantly better than that of SVM. It should be noted that McFIS attains this performance employing about 90% of the training samples and 250 rules alone. Similarly, for KTH data set it could be seen that, McFIS outperforms SVM. The poor performance of SVM is due to the fact that it solves multi-category classification problems as several binary classification problems, whereas the action recognition problem considered here is a multi-class classification problem.

5.2.4.3 Comparison of Sub-sequence Level Recognition Performance

Next, we compare the action recognition results of the proposed approach with some of the state-of-the-art methods for Weizmann and KTH data sets. In this section simple optical flow based features are employed whereas [69] and [68] employ silhouette based global level information. These information are prone to noise and are difficult to extract. In [236] bag of words model is employed where, the motion descriptors are obtained after tracking and stabilizing the persons in the video. Authors in [25] employed quantized spatio-temporal features which employs figure-centric information about the local spatio-temporal distribution to compute statistics of pairwise co-occurring visual words within its neighbourhoods.

The comparison for Weizmann data set is given in table 5.7 and for KTH data set in table 5.8. It could be seen that, the performance of the proposed method is better

Table 5.7: Performance Comparison of McFIS-classifier with State-of-the-art Action Recognition System for Weizmann Data Set

Approaches	Optical flow		Silhouette	Space-time shapes	Kinematic features
Algorithm	McFIS	SVM	Ref.[69]	Ref.[68]	Ref.[7]
Test Accuracy	99.46%	87.34%	96.74%	97.83%	95.75%

Table 5.8: Performance Comparison of McFIS-classifier with State-of-the-art Action Recognition System for KTH Data Set

Approaches	Optical flow		Silhouette	Space-time shapes	Kinematic features	QLSTF ^a
Algorithm	McFIS	SVM	Ref.[236]	Ref.[184]	Ref.[7]	Ref.[25]
Test Accuracy	95.4%	63.8%	91.2%	71.7 %	87.7%	96.16%

(a)Quantized local spatio-temporal features

than the state-of-the-art approaches reported in literature.

In the next section, a 3D action recognition system based on McFIS-classifier is proposed. In order to represent 3D actions, a 3D optical flow based feature representation technique is employed. The performance of such an action recognition system is evaluated on a proprietary data set.

5.3 Human Action Recognition based on 3D Video Sequence

5.3.1 Motivation

Human action recognition is one of the widely employed video analytics in a number of real-world applications such as video surveillance, content-based video search, health care, etc. Until recently, the applications of action recognition were restricted to video sequence captured by 2D (RGB) cameras [249, 196, 198, 69]. In here, the video is a temporal sequence of 2D frames of RGB images. Various research have been conducted in the domain of action recognition based on 2D video sequence. A brief review is available in 5.2.2. However, with the advancement in imaging techniques, such as Microsoft Kinect and ASUS Xtion Pro Live, it became feasible to capture depth maps in addition to RGB sequence, in real time. The depth map information provides information regarding body

shape, occlusion, etc., which helps in identifying view point variation. In most of the commonly employed action recognition techniques, the actions are performed parallel to the image plane, in order to capture the dynamics with less ambiguity. This results in the developed action recognition system being view-point dependent. In the context of human action recognition, 3D help in differentiating actions having similar 2D projections from a single view. This finding has motivated researchers to study action recognition based on 3D video sequences. A review is provided in Section 5.3.2.

In the previous section, it was shown that optical flow based features can capture the apparent movement in object, naturally, and can moreover represent information hierarchically from local pixel level to global object level. In this section, such an optical flow based features have been extended to represent information from 3D video sequence. In order to extract 3D optical flow based features, 2D optical flow features are combined with depth motion vectors calculated using 2D optical flow technique on depth images. A detailed description on the 3D optical flow based feature extraction is provided in Section 5.3.3.

In this chapter, we employ a Meta-Cognitive Neuro-Fuzzy Inference System based Classifier (McFIS-classifier) for efficiently mapping actions to their corresponding 3D video sequence. The performance of such a 3D action recognition system is evaluated on a proprietary data set consisting of eight subjects, and each of them performing eight different actions. The performance evaluated based on 10-fold cross validation study and performance comparison against standard support vector machines and extreme learning machine classifier indicates improved performance of McFIS-classifier based action recognition system.

5.3.2 Background and Related Work on 3D Action Recognition

Similar to 2D action recognition, the feature extraction for 3D action recognition assumes a video sequence to consist of only one action. Accordingly, various works have been done in this domain. In [116] an action graph is employed to model the dynamics of actions based on sequences of depth maps. Here, sampled 3D representative points from the contours of depth maps of a body surface is projected onto three orthogonal Cartesian planes. However, the sampling scheme is view dependent. In [245], 3D skeletal

joint locations from Kinect depth maps are mapped to a spherical coordinate system and histogram of modified 3D joint positions is used to achieve view-invariant posture representation. In [189] a method to predict 3D positions of body joints from a single depth image from Kinect is proposed. A random transformation on depth silhouettes was employed in [92] to recognize human home activities. However, this method is also view dependent. In [208], skelton data provided by Prime Sense from RGBD data of Kinect are employed to extract features. It employs a supervised learning approach to inter activities from RGB and depth images from Kinect.

In the next section, 3D optical flow based feature extraction technique is described.

5.3.3 3D Optical Flow Estimation

First step in 3D optical flow feature extraction is silhouette extraction. In order to extract silhouette, first the RGB and depth video segments are calibrated to map the pixel locations in both these frames. The depth image is appropriately normalized by representing it as 8 bit data, and this normalized data is used to extract silhouette.

Optical flow feature extraction consists of first extracting 2D optical flow features between the consecutive images for the silhouette region. 2D optical flow calculation is a well known problem and many methods are available for calculation of flow vectors. In this work, we employ Lucas and Kanade's method [124] for feature extraction. The feature extraction procedure for 2D image sequence has been described in detail in Section 5.2.3.1.

The depth image obtained from the depth camera is used to calculate motion vectors in the Z direction. The depth motion vectors are calculated using the 2D optical flow technique on depth images, which involves subtracting the depth value of the same point in two consecutive depth frames.

From the 3D optical flow vector, the optical flow based features are extracted as described in [21]. First a bounding box is obtained by accumulating the motion vectors for a sequence of N frames. This bounding box is hierarchically divided into 54 windows placed symmetrically with respect to the silhouette center. The box is divided into 6×6 windows, 3×3 windows, 2×2 windows, 2×1 windows, 1×2 windows and 1×1 window of equal size. From the windows, average motion of each window is calculated by averaging

the non-zero motion vectors of all three dimensions in each window. Thereby generating a feature vector of length 162 (54×3) for every frame.

Since the average motion of a single frame would not contain significant amount of information to represent an action, the average motion of N frames are employed to represent an action.

5.3.4 Performance Evaluation of 3D Action Recognition System

In this study, we employ a proprietary 3D action data set. In order to generate the 3D action action, eight subjects were considered and each of them were asked to perform eight different actions, namely, swimming, bending, waving, kicking, bowling, jumping, boxing, and stretching. A snapshot of some of the actions from our database is shown in Figure 5.6. Each action was performed by a subject atleast three times. In total 3241 action sequence were obtained. The performance evaluation is conducted based on 10-fold cross validation, wherein, 75% samples were employed in developing the classifier and the remaining 25% samples are employed as test samples. The performance of the algorithms are compared using the following measures:

- Average classification efficiency (η_a):

$$\eta_a = \frac{1}{n} \sum_{l=1}^c \frac{q_l}{N_l} \times 100\% \quad (5.6)$$

where, q_l is the total number of correctly classified samples n the training/testing data set.

- Overall classification efficiency (η_o):

$$\eta_o = \frac{\sum_{l=1}^c q_l}{N} \times 100\% \quad (5.7)$$

The performance of McFIS on 3D human action recognition database has been conducted in MATLAB R2011b on a Unix system with 16GB memory. The performance of McFIS is compared with optimized results of support vector machine (SVM) [39] with Gaussian kernel and extreme learning machines (ELM) [84] with radial basis function kernel. The parameters of McFIS and SVM were optimized employing grid search based

(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.6: A Snapshot of Sample 2D (Bend, Swim and Kick) Actions (a) - (c) and the Corresponding Depth Images (d) - (f) Used in 3D Action Recognition

technique, whereas the results for ELM were chosen based on constructive-destructive parameter selection approach.

Table 5.9 presents the training and testing efficiencies of the three classifiers considered for 10-fold cross validation study. The results of SVM and ELM are obtained by optimizing the parameters. From the performance results, it can be observed that McFIS outperforms SVM and ELM classifiers by 2% to 11% respectively. The meta-cognitive components in McFIS, along with the class-specific knowledge based criteria and self-adaptive thresholds have helped the network learn the samples, efficiently.

Table 5.9: Performance Comparison of McFIS-classifier based 3D Action Recognition System with SVM on 10-Fold Cross Validation Data Set

Classifiers	K	Training		Testing	
		η_o	η_a	η_o	η_a
SVM	1589.1	96.9 ± 2	96.85 ± 0.4	96.8 ± 0.54	96.4 ± 1.3
ELM	90	93.1 ± 0.6	88.42 ± 1.37	93.2 ± 0.5	88.97 ± 1.2
McFIS	119.2	98.41 ± 1.45	97.40 ± 2.32	98.39 ± 1.6	98.12 ± 1.79

5.4 Summary

In this chapter, McFIS-classifier based human action recognition system is proposed. The action recognition system classifies human actions based on the optical flow based features extracted from RGB video sequence. Optical flow based features between two image planes is employed as it can capture the apparent movement in object naturally and can further represent the movement hierarchically from local pixel level to global object level. Meta-cognitive learning algorithm decides on *what-to-learn*, *when-to-learn* and *how-to-learn* to efficiently and accurately find the decision boundary separating different classes based on the optical flow based features. The performance of the proposed system is compared against standard support vector machine and other state-of-the art action recognition systems available in literature. The performance comparison indicates superior performance of McFIS-classifier in recognizing human actions.

The proposed McFIS-classifier based human action recognition system is further extended to solve 3D action recognition. In order to represent 3D action, 2D optical flow based feature extracted from RGB video is combined with depth flow information, which

in turn is obtained by extracting optical flow based feature from depth map information. This renders the action representation view invariant. The performance of 3D action recognition system is evaluated on a proprietary action recognition data set. A 10-fold cross validation study indicates superior performance of McFIS-classifier based action recognition system in comparison to support vector machines and extreme learning machines.

Chapter 6

Meta-Cognitive Neuro-Fuzzy Inference System based Classifier for Facial Emotion Recognition

A Meta-Cognitive Neuro-Fuzzy Inference System based classifier (McFIS-classifier) for action recognition was proposed in the previous chapter. The developed system was employed for efficient recognition of human actions from 2D as well as 3D video sequences. In this chapter, a facial emotion recognition system is proposed employing McFIS-classifier. We evaluate the emotion recognition ability of McFIS by conducting a 10-fold cross validation study. We also evaluate the ability of various feature extraction techniques in representing human emotions efficiently. Facial expression recognition depends on how the facial features are encoded. In this chapter, we study the effect of different features and the robustness of McFIS-classifier in facial emotion recognition in generalizing emotion features. Two kinds of study are conducted: person independent study and database independent study. To conduct these experiments, we employ five different feature extraction techniques, pixel based, local binary pattern based, scale invariant feature transform based, Gabor filter based and curvelet based features from two well-known benchmark databases. The aim of McFIS-classifier is to approximate the functional relationship between the emotion classes and the corresponding facial features.

Figure 6.1: Overview of McFIS Based Emotion Recognition System

6.1 Overview

In this section, an overview of McFIS-classifier based emotion recognition system is proposed. The Fig. 6.1 gives an overview of the proposed facial emotion recognition system. Emotion recognition based on McFIS-classifier consists of two parts: training and testing. During the training process, the raw image is pre-processed and face region extracted. This is followed by background segmentation and extraction of emotion feature descriptors. The next step involves training. McFIS-classifier learns the functional relationship between the annotated facial features and the emotions they represent. The rules representing the functional relationships are stored in the rule base.

During testing, similar to the training phase, raw image is pre-processed, face extracted, segmented and emotion features are extracted. McFIS-classifier employs the rules in the rule base to predict the emotions based on the extracted facial features. Various feature extraction techniques are available in the literature of computer vision. A

review about the same is provided in Sections 6.3. The extracted features are processed by McFIS-classifier to predict the corresponding emotions.

6.2 Motivation

Facial expression is a visible measure of a person's affect and it can be used to study one's character, personality, intentions and emotions [47]. Various works dealing with the study of human emotion based on facial features have been reported in literature. In [51, 50] authors have shown that facial expressions related to emotions are universal in nature and constant across cultures. In [52], authors have shown that certain emotions, including happiness, sadness, anger, surprise, disgust and fear are expressed the same way across cultures and are 'discriminable within any one literate culture'. Apart from these six basic emotions, other emotions can also be depicted by human face [153].

In recent times, with the advances in the fields such as robotics, computer vision and human computer interactions, computer scientists have started showing significant interest in the study of facial expressions, with the aim of automating the task of emotion recognition based on these facial expression. The task of automatic facial expression analysis is an important and challenging problem in various activities such as, operator fatigue detection, mood detection, suspicious person detection in public places, etc. Facial emotion recognition involves two steps, extraction of features which can capture the facial features efficiently and an emotion recognition system, which can recognize various emotions based on the extracted features. In literature various features have been employed for efficient representation of facial features, such as pixel based features [106], local binary pattern [148, 4], curvelet transform based approach [217, 132], scale invariant feature transform [122, 251] and Gabor filter based approach [246]. Each of these approaches have been shown to be efficient in representing facial features for different applications. Various classifiers have also been employed for efficient classification of human emotions based on these features. A brief review of major works in the domain of facial emotion recognition is presented in Section 6.3.

In many real-world problems, such as emotion recognition, the data for training almost always arrive sequentially and are massive in size. Hence, we require algorithms

which can learn from stream of data in an online fashion. Artificial neural networks [177, 178], fuzzy inference systems [176, 61] with their respective predictability and interpretability properties have been shown to be able to handle such sequential data. Neuro-fuzzy inference systems [196, 169, 121] which are a combination of artificial neural networks and fuzzy inference systems can achieve the twin requirements of predictability and interpretability. Moreover, they have been shown to be able to deal with online sequential data, efficiently. As a result, they have been employed in various applications (A review has been provided in Chapter 2).

In this chapter, we employ the recently propose Meta-Cognitive Neuro-Fuzzy Inference System based classifier (McFIS-classifier) for the task of human emotion recognition. The aim of such an emotion recognition system is to find the decision boundary separating various emotions based on different facial features extracted. In order to study the effect of different feature extraction techniques on facial emotion recognition with McFIS-classifier, five different features are employed in this study, pixel based, local binary pattern based, Gabor filter based, curvelet based and scale invariant feature transform based feature. Pixel are computationally inexpensive descriptor which can be extracted directly from an image. Hence, they are used as base feature in this study. Local binary patterns are simple texture based features which are extracted from local patches of an image. Although it is not as simple as pixel based features, it is computationally inexpensive to extract and has been shown to be able to represent information accurately. Gabor filter based features are extracted by blurring and detecting edges at different image orientations, whereas scale invariant feature transform does the same at different scale, rather than orientation. However, these techniques employ Gaussian or Laplacian of Gaussian operations, and hence computationally expensive to calculate. Curvelet features based approach employs multi-resolution, band pass and multi-orientation based approach to extract edge information at multiple scales, orientation, and resolution. This feature extraction technique combines the advantages of spatial as well as orientation shifts. However, the cost of extraction is significantly higher.

In this study, we conduct a 10-fold cross validation study on these features extracted from two well known data bases: Japanese female facial expression database [135] and Taiwanese facial expression identification database [36]. Moreover, we also conduct a person

independent study with database to study the generalization ability of McFIS-classifier based emotion recognition system in approximating the decision boundary, efficiently. Next, the performance study is conducted to study the database independent emotion recognition ability of McFIS-classifier. Here, the network is trained on features extracted from one database and is tested on the same feature extracted from other databases. The comparative study on various features and database independent study clearly indicates the superior performance of McFIS-classifier.

Rest of the chapter is organized as follows. In Section 6.3, a review of major works in the domain of facial emotion recognition is presented. In Section 6.4, we describe the feature extraction procedure of the different features employed in this study. Section 6.5 gives a performance evaluation of McFIS-classifier on emotion recognition.

6.3 Review of Major Works in Facial Emotion Recognition

In any facial expression recognition system, there are three steps, face tracking and detection, feature extraction and expression classification [222]. In this study it is assumed that the face has been detected. In literature, various approaches have been employed for extracting representative facial features and recognizing emotions based on these features.

One of the first works in the field of automatic facial emotion recognition was published in [213]. A comprehensive approach for facial emotion representation is the facial action coding system proposed in [48], which employed human operators to observe face and give analysis. However such visual observation cannot be considered as an exact science as the judgement may vary drastically over observers. The recent research works [134] indicates that the facial expressions do not occur in separate locations, as suggested above, but as a combined movement. Various methods which can automatically identify emotions from face images have been proposed in literature. In [32] authors have employed linear discriminant analysis to classify principal component features of the face images, whereas in [30], authors have employed independent component analysis based approach for expression recognition based on Gabor wavelets. Authors are also employing general discriminant analysis for extracting representative features from facial images [188].

In [248, 53], optical flow based features were employed for modeling facial features, based on the assumption that all expressions contain different motion information that can be separated using motion features. They modeled the muscular activities or estimate the movement in the facial feature points. One of the disadvantages of optical flow based features is their sensitivity to lighting variation and disturbance caused due to non-rigid motions. Moreover, they are also sensitive to inaccurate image representation and discontinuous motions. Recently local binary patterns have been employed as effective appearance features for facial image analysis [70, 186]. In [186], authors have compared local binary patterns with Gabor wavelet based features and illustrated the effectiveness of the same on low resolution images. In [194], a discriminative scale invariant feature transform is employed for constructing representative facial keypoints, whereas, in [85], a speeded-up robust features is employed for generating keypoint descriptors of face images. In both the above algorithms, a Kullback Leibler and weighted majority voting classifier is employed for classification. A Gabor filter based facial feature extraction procedure is employed in [111, 150]. Here, features obtained by employing Gabor filter bank with five frequencies and eight orientations are used. A KNN classifier on reduced feature set obtained by employing principal component analysis is used in the study.

A facial feature representation technique based on curvelet transform is presented in [217]. The amount of curvelets generated is reduced by sorting and selecting the most important features. A support vector machine based algorithm is employed for classifying the expressions. In [174], curvelet transform is combined with local binary patterns for efficient facial feature extraction. Local binary patterns based features are extracted from curvelet transform applied to image at a particular orientation and scale. A nearest neighbor classifier is employed to classify the emotions.

To learn the decision boundary separating various emotions, various approaches have been proposed in literature. In [221], artificial neural network based classifier was employed to extract Gabor wavelets based facial action units. In [110], a support vector machine based approach was combined with geometric deformation features for efficient recognition of facial expression. In literature various classifiers based on the combination of artificial neural networks and fuzzy inference systems, called neuro-fuzzy inference systems [169, 121] have been proposed. These neuro-fuzzy inference systems combine

the predictability and interpretability properties of neural networks and fuzzy inference systems, respectively. However, most of the neuro-fuzzy inference systems employed in literature do not efficiently self-regulate their knowledge. In this thesis, we presented a meta-cognitive neuro-fuzzy inference system which could self-regulate its knowledge and decide on selecting *what-to-learn*, *when-to-learn* and *how-to-learn*, efficiently. In this chapter, we employ such a meta-cognitive neuro-fuzzy inference system based classifier for recognizing emotions from facial expressions.

In the next section, the different feature extraction procedure employed in this chapter are described.

6.4 Feature Extraction Procedure

The problem of facial emotion recognition involves two steps: extraction of facial emotion features and classification of emotions based on these features. In this chapter we employ five different feature extraction techniques, pixel based features, local binary patterns based features, curvelet based features, scale invariant feature transform and Gabor filter based features. These feature extraction techniques are described next.

In each of these methods, it is assumed that the image has been normalized and cropped to contain only the face region (from forehead to chin), as shown in Fig. 6.2. The Fig. 6.2(a) shows an original image from JAFFE database with image id. TM.NE2.178. The corresponding cropped image employed in this study is as shown in Fig. 6.2(b). Such a cropping technique helps to eliminate any discrepancies arising due to background variations or person appearance variations, such as hairstyle.

6.4.1 Pixel based Feature Extraction

In this method, the pixel values are directly used to form the feature vector. In order to extract pixel based feature vector, the cropped images have been resized to 20×15 pixels. The resulting 300 pixels are employed as the feature vector for each image.

6.4.2 Local Binary Pattern based Feature Extraction

Local Binary Pattern (LBP) is a non-parametric and computationally simple image descriptor. It converts an image into an array or image of integer labels describing small-

Figure 6.2: Image Cropping Technique Employed for Emotion Recognition Study

Figure 6.3: Example for Local Binary Pattern Feature Extraction Process for Emotion Recognition Using Image TM.NE2.178 from JAFFE Database [135]

scale appearance of image. The LBP operator is based on the assumption that texture has a pattern and its strength associated with it. The original LBP operator labels the pixels of an image by thresholding a 3×3 neighborhood of each pixel with the center value and considering the results as a binary number, of which the corresponding decimal number is used for labeling. In recent times, an extension of LBP has been proposed which uses neighborhoods of different sizes. Here, neighborhoods are defined as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled, and the sample points that do not fall in the pixels are expressed using bilinear interpolation, thus allowing any radius and number of sampling points in the neighborhood. An example LBP extraction from an extended LBP operator of radius one and eight points is shown in Fig. 6.3.

In this study, we consider two sets of LBP parameters: radius six and two sampling points, and radius eight and four sampling points. The original image is resized to

(a) Cropped Image of TM.NE2.178 in JAFFE Database [135]

(b) Corresponding SIFT

Figure 6.4: Scale Invariant Feature Transform Feature Extraction Process for Emotion Recognition Using Image TM.NE2.178 from JAFFE Database [135]

dimension 20×15 and resulting feature vector for each pixel and for each LBP parameters are concatenated to form a final feature vector of size 600 ($20 \times 15 \times 2$).

6.4.3 Scale Invariant Feature Transform based Feature Extraction

Scale Invariant Feature Transform (SIFT) is a computer vision algorithm to detect and describe local features in images. The main advantages of SIFT features are its scale, rotation, illumination and viewpoint invariance. In order to extract SIFT based features, first step is to construct scale space by employing Laplacian of Gaussian operator. In the next step, difference of Gaussian is calculated. Then, dense SIFT descriptors are computed for each pixel. For a given image of size $w \times h$ pixels, a 3D SIFT image of dimension $w \times h \times 128$ pixels is obtained. To visualize SIFT images, 128D SIFT vector is projected onto the 3D RGB space, by mapping the top three principal components of SIFT to the principal components of RGB. As a result of this the feature vector was obtained as $26 \times 26 \times 3$. This was subsampled in steps of three to obtain a final feature vector length of 676.

An example of SIFT based feature extraction results on an example image is shown in Fig. 6.4. The Fig. 6.4(a) shows the original image with id TM.NE2.178 from JAFFE database and the Fig. 6.7(b) shows the corresponding SIFT based features.

6.4.4 Gabor Filter based Feature Extraction

Gabor filter is a linear filter which has been found to be a good choice for texture representation and discrimination. The core of Gabor filter based feature extraction technique is the Gabor filter function. It is a linear filter represented as a Gaussian kernel function modulated by a sinusoidal plane wave, in the spatial domain. It is given as,

$$\begin{aligned} Gabor(x, y) &= \frac{f^2}{\pi\gamma\eta} e^{-\left(\frac{f^2}{\gamma^2}x'^2 + \frac{f^2}{\eta^2}y'^2\right)} e^{j2\pi fx'} \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned}$$

where, f is the central frequency of filter, θ , the rotation angle, γ , η are the Gaussian blurring factors. The cosine component may be construed as the low frequency part (directional blur operator), and the sine part as the high frequency component (directional edge detector). The Gabor filter bank is created by varying the frequency and orientation of the Gabor filter. In this study, a discrete set of Gabor kernels with three spatial frequencies (with wave-numbers $k = \pi/4, \pi/8$ and $\pi/16$) and six distinct orientations from 0° to 150° , with an interval of 30° steps. Thus a filter bank with 18 different Gabor filters, each of size 16×16 is created. The cropped image is resized to 64×64 pixels. These pixels are then divided into non-overlapping patches of size 16×16 , as shown in Fig. 6.5. The sine and cosine feature vectors obtained for each image are appended and used as a new feature vector. The feature vector is of length 576 ($16 \times 18 \times 2$).

6.4.5 Curvelet based Feature Extraction

Curvelet transform is a technique of multi-resolution, band pass and direction analysis which can capture image edge information more accurately than conventional spectral methods such as Gabor filters. It has good time-frequency localization characteristics and can characterize the facial local information effectively. Furthermore, the curvelets have been shown to have the sparse property compared to wavelets in the sense that curvelets decomposition can approximate the original image better with less non-zero coefficients [259].

Figure 6.5: The Non-Overlapping Patches in Image TM.NE2.178 from JAFFE Database [135] for Extraction of Gabor Filter Based Features

Figure 6.6: Scale and Angle Representation of Curvelet Transform Used in Feature Extraction for Emotion Recognition

The output processing of Curvelet transform is shown in Fig. 6.6. The figure shows the scale and angle segmentations in curvelet transform. Each layer (surrounded by rectangular box) segments the image into different scales and frequencies. The inner most layer is a coefficient matrix corresponding to the low frequency of dimension 32×32 , and the outer most layer is coefficient matrix of high frequency of $n \times n$, n being the image dimension. The layers between the inner most and the outer most layer represents coefficient matrix of image different scale and extracted at different frequencies. Every layer is further divided into four round angles, each of which is divided into several small directions. There is a coefficient matrix in every sub-angle. The coefficients of low frequency can represent the essential characteristics of a human face, accurately.

In this study, the inner most facial characteristics are employed for emotion represen-

(a) Cropped Image of TM.NE2.178 in JAFFE Database [135] (b) Corresponding Curvelet Coefficients

Figure 6.7: Curvelet Features Extracted from an image TM.NE2.178 from JAFFE database [135] for Emotion Recognition

tation. The feature vector is of length 1024 (32×32). The extracted curvelet features for an example image is shown in Fig. 6.7. The Fig. 6.7(a) shows the original image with id TM.NE2.178 from JAFFE database and the Fig. 6.7(b) shows the corresponding curvelet coefficients.

In the next section, the performance of each of these feature extraction techniques shall be studied.

6.5 Performance Study

In the previous section, five different feature extraction techniques were discussed. In this section, the performance of each of these features are evaluated on McFIS-classifier based emotion recognition system. In order to evaluate the generalization ability of McFIS-classifier and the ability of each of these features to capture distinct facial features, a person independent study is performed. Here, the recognition system is trained on all subjects, except one and tested on the remaining one individual. Further, this concept has been extended to perform a database independent emotion recognition. Here, the emotion recognition system is trained on one database and is tested on similar features extracted from an entirely different database. The performance of McFIS-classifier on each of these problems is compared against support vector machine (SVM) [39] based classifier with Gaussian kernel, on emotion recognition problems. All the experiments

Figure 6.8: Sample Facial Images from JAFFE [135] Database for Emotion Recognition

reported in this study are conducted in Matlab 2011R runtime environment on Unix based operating system with 16Gb memory. The tunable parameters of McFIS and SVM were optimized employing grid search based technique.

In this section, first we shall describe the databases employed in the study. Then, we shall present the results of 10-fold cross-validation study conducted on each of these databases, followed by person independent and database independent study.

6.5.1 Dataset Description

In this study, we employ two datasets, Japanese Female Facial Expression Database (JAFFE) [135] and Taiwanese Facial Expression Identification Database (TFEID) [36]. Sample images of JAFFE database is shown in Fig. 6.8.

The JAFFE database consists of 213 images of seven frontal shots of facial expressions (six basic facial expressions and one neutral) posed by ten Japanese female models. The images were labeled with the predominant emotions as rated by 60 Japanese subjects. The TFEID database is a growing database with new images in different angle and pose are being added. However, in this study, only the frontal facial images are considered. Each individual facial expression in TFEID have been classified into eight different emotions. During cross-validation and person independent study 336 images with eight emotions are considered, whereas, during database independent study, 268 images corresponding to seven common emotions to JAFFE database is only considered.

6.5.2 10-Fold Cross Validation Study

In this section, we shall present the results for 10-fold cross validation study conducted on JAFFE database and TFEI database. In order to extract 10-fold cross-validation data set for JAFFE, one image per person per emotion is employed for testing and rest of the images are employed for training. However, since TFEID does not contain equal number of images per person per emotion, a 10-fold cross validation data set is created such that 66% of images per emotion are employed for training and rest 33% images per emotion are employed for testing.

Table 6.1: Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM on Five Features extracted from JAFFE 10-Fold Cross Validation Database

Method	McFIS-Classifier			SVM		
	No. of Rules	Test		No. of S.V ^a	Test	
		Mean	S.D.		Mean	S.D.
Pixel	78	85.29	3.92	139	80.81	3.51
LBP	84	83.29	4.21	139	76.71	4.15
Gabor	85	82.34	5.25	131	62.24	5.79
SIFT	83	82.14	3.94	131	66.29	4.05
Curvelet	83	84.85	4.69	138	77.72	6.92

(a) Support vectors

Table 6.2: Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM on Five Features extracted from TFEID 10-Fold Cross Validation Database

Method	McFIS-Classifier			SVM		
	No. of Rules	Test		No. of S.V ^a	Test	
		Mean	S.D.		Mean	S.D.
Pixel	78	84.30	2.61	138	77.95	6.85
LBP	76	80.31	2.95	140	73.85	3.33
Gabor	75	84.15	3.91	133	79.79	4.04
SIFT	86	82.40	1.71	137	79.50	1.59
Curvelet	66	90	3.01	139	81.98	2.91

(a) Support vectors

In Tables 6.1 and 6.2, we provide results for 10-fold cross-validation studies on JAFFE and TFEI databases, respectively. The tables provides the number of rules employed,

training mean and standard deviation (S.D.), testing mean and standard deviation for McFIS-Classifer as well as SVM, for all the five features considered. It could be observed from the tables that McFIS-classifier outperforms SVM by 2% - 8% employing significantly lesser number of rules, for both JAFFE and TFEID datasets.

In JAFFE dataset, pixel based features help attain the best performance, followed by curvelet based features. In TFEID dataset, curvelet based features helps attain better recognition. It could be observed that in general, curvelet based features can capture the the underlying decision boundary efficiently in 10-fold cross validation study. This might be due to the multi-scale, multi-orientation and multi-resolution features extracted by curvelet based features. In order to evaluate the generalization ability of the different features and McFIS-classifier, person independent emotion recognition study is conduction in the next section.

6.5.3 Person Independent Study

In [52] it has been argued that human emotions are ‘discriminable within one literate culture’. As a result, any emotion recognition system trained on a particular individual should be able to generalize well and be able to recognize emotions of different individual of the same culture. In this section, we study the performance of different features in generalizing emotions within the same databases. We perform a person independent study on JAFFE and TFEI databases, separately, wherein images related to one person is employed for testing while the network is trained on remaining images. In TFEID, certain people do not contain some features. Here, images related to only those people are selected, who have at least one representative image for each emotion.

In Tables 6.3 and 6.4, we provide results for person independent studies on JAFFE and TFEI databases, respectively. The tables provides the number of rules employed, training mean and standard deviation (Std.), testing mean and standard deviation for McFIS-Classifer as well as SVM, for all the five features considered. From the Tables 6.3, 6.4, one could notice that McFIS-classifier outperforms SVM by a significant margin. It achieved this best performance employing less than 50% of rules as SVM.

The standard deviation reported in the Tables 6.3, 6.4 clearly highlight the generalization ability of McFIS. For both the problems considered, McFIS-classifier attains

Table 6.3: Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM on Five Features extracted from JAFFE Person Independent Database

Method	McFIS-Classifier			SVM		
	No. of Rules	Test		No. of S.V ^a	Test	
		Mean	S.D.		Mean	S.D.
Pixel	95	72.86	0.1	223	66.55	0.13
LBP	95	60.41	0.11	261	45.24	0.14
Gabor	95	73.21	0.05	255	64.21	0.01
SIFT	96	76	0.06	142	68.2	0.31
Curvelet	95	69.27	0.08	261	64.22	0.81

(a) Support vectors

Table 6.4: Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM on Five Features extracted from TFEID Person Independent Database

Method	McFIS-Classifier			SVM		
	No. of Rules	Test		No. of S.V ^a	Test	
		Mean	S.D.		Mean	S.D.
Pixel	113	96.29	0.08	220	78.57	0.32
LBP	104	95.05	0.04	234	78.35	0.32
Gabor	98	95.15	0.06	260	86.23	0.02
SIFT	112	98.04	0.09	140	79.52	0.31
Curvelet	111	93.09	0.09	245	73.65	0.57

(a) Support vectors

a standard deviation less than 0.1 implying that irrespective of the person chosen, the proposed algorithm is able to learn the underlying decision boundary efficiently. Moreover, it could be observed that SIFT based features help McFIS and SVM attain their best person independent performance. The use of different scale and orientation of SIFT based features would have helped SIFT based features in representing dataset invariant features.

6.5.4 Database Independent Study

In literature [51, 50] it has been shown that basic emotions are represented similarly across different cultures. As such, any machine learning algorithm aiming at recognizing emotions needs to be able to handle minor variations in emotion representation across cultures and be able to recognize similar emotion in different culture, accurately. In order to study this aspect of human emotion recognition, we propose to conduct a database independent emotion recognition study in this section. In order to conduct this study, the emotion recognition system is trained on one database and is tested on other database. This study considers only samples representing the seven common emotions in both JAFFE and TFEI databases.

In Tables 6.5 and 6.6, we provide results for database independent studies on JAFFE and TFEI databases, respectively. The tables provides the number of rules employed, training accuracy and testing accuracy for McFIS-Classifier as well as SVM, for all the five features considered.

Table 6.5: Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM Trained on Five Features extracted from JAFFE Database and Tested on TFEID

Method	McFIS-Classifier			SVM		
	No. of Rules	Train	Test	No. of S.V ^a	Train	Test
		Accuracy	Accuracy		Accuracy	Accuracy
Pixel	32	79.84	42.05	143	100	27.51
LBP	105	95.294	40.3	213	100	29.38
Gabor	139	97.18	48.88	213	100	29.43
SIFT	126	95.31	51.31	200	100	34.52
Curvelet	102	94.84	49.5	213	100	33.53

(a) Support vectors

Table 6.6: Performance Comparison of McFIS-classifier based Emotion Recognition System with SVM Trained on Five Features extracted from TFEID Database and Tested on JAFFE

Method	McFIS-Classifier			SVM		
		Train	Test	No. of S.V ^a	Train	Test
		Accuracy	Accuracy		Accuracy	Accuracy
Pixel	128	87.36	43.23	143	100	29.43
LBP	96	93.66	42.5	268	100	30.15
Gabor	185	88.43	50.7	268	100	35.15
SIFT	188	89.52	51.02	268	100	30.55
Curvelet	97	95.6	42.8	268	100	29.43

(a) Support vectors

The Tables 6.5, and 6.6 provide a performance comparison of McFIS-classifier and SVM on recognizing emotion independent of database, for various features considered. The tables provide the number of rules/ support vectors employed, training overall accuracy and testing overall accuracy upon employing each feature. In general, McFIS-classifier outperforms SVM in cross-database study. However, both the algorithms achieve significantly less accuracy during testing.

All features except pixel based features are able to capture the underlying similarity in facial emotion, across datasets. In the case of JAFFE, and TFEID, SIFT based features help attain the best performance. The use of different scale and orientation of SIFT based features would have helped SIFT based features in representing dataset invariant features.

6.6 Summary

In this chapter, McFIS-classifier is employed for facial emotion recognition. Three different studies are conducted: cross-validation study, person independent study and database independent study. The aim of the first study is to show the general emotion recognition ability of McFIS, whereas, the last two studies aim to highlight the generalization ability of McFIS in recognizing emotions, independent of person and further, independent of database. Moreover, we also compare the ability of five different features in representing emotions accurately and generically. The feature descriptors chosen as pixel based

descriptor, texture descriptor, and edge descriptors. Pixel is a simple image descriptor, whereas local binary patterns are texture based image descriptors. Gabor filters, employ blurred edges for representing images, and scale invariant feature transform attains the same at multiple scales and orientation. Curvelet based feature extracts image edges at multiple scales, orientation and resolution.

The performance comparison with SVM clearly highlights the emotion recognition ability of McFIS-classifier. The performance study on McFIS-classifier shows that during cross-validation, all the descriptors perform equally well with curvelet based features performing the best. However, during person independent and database independent study SIFT attains the best performance. It should be noted that during database independent study, none of the approaches attain a satisfactory performance. This clearly indicates the need for a better feature representation technique which can generalize the emotion irrespective of the database.

Chapter 7

Conclusions and Future Direction

7.1 Thesis Summary

The aim of this thesis is to develop a meta-cognitive sequential learning algorithm for a neuro-fuzzy inference system for efficiently generalizing functional relationship between input and output. The major contributions could be summarized as given below:

- Development of a self-regulatory learning mechanism, which enables a learner to decide on *what-to-learn*, *when-to-learn* and *how-to-learn*.
- Development of meta-cognitive self-regulatory learning mechanism for neuro-fuzzy inference systems, where the meta-cognitive component monitors the knowledge in the neuro-fuzzy inference system and efficiently controls the learning by deciding on *what-to-learn*, *when-to-learn* and *how-to-learn*.
- Extension of meta-cognitive sequential learning algorithm for neuro-fuzzy inference systems to efficiently solve classification problems. The developed learning algorithm employs class-specific information, while learning as well for deciding on *what-to-learn*, *when-to-learn* and *how-to-learn*.
- Development of technique to improve rule exploitation and generalization in neuro-fuzzy inference systems employing Gaussian membership functions.
- Performance evaluation of two practical problems of human action recognition and facial emotion recognition.

We shall now describe each of the contributions, in detail, below.

7.1.1 Self-Regulatory Learning Mechanism

In this thesis, a self-regulatory learning mechanism has been developed for a neuro-fuzzy inference system. Self-regulated learning refers to the ability of a learner to efficiently decide on *what-to-learn*, *when-to-learn* and *how-to-learn*. The knowledge of *what-to-learn* requires the ability to judge and identify a specific piece of knowledge that is lacking. The knowledge of *when-to-learn* requires the ability to judge when to start a learning process and when to stop it. The *how-to-learn* ability requires the knowledge of the available learning strategies available and also the knowledge to choose the best strategy available to learn what is required. The self-regulatory learning mechanism controls the learning process by initiating one of the following strategies:

- *Sample deletion strategy*: Employing this strategy, the learning mechanism removes the samples with insignificant knowledge from the learning system, thereby reducing the training time significantly. This strategy realizes the *what-to-learn* component of self-regulation.
- *Sample reserve strategy*: This strategy helps in deciding which sample to learn at a given time, thus realizing *when-to-learn* component of self-regulation. A sample with less knowledge is pushed to the end of the data stream to be learnt at a later stage.
- *Sample learning strategy*: The learning mechanism employs this strategy to gain the knowledge stored in a sample, there by realizing *how-to-learn* component of self-regulation. The sample learning strategy works by either adding a new rule to the network, updating the parameters of the network or removing redundant/under performing rules from the network.

Employing the above three strategies during the training of neuro-fuzzy inference systems, one could efficiently accomplish sample selection and model selection, simultaneously. It must be noted that in many real-world problems, the data for training the inference system are not available *apriori*. As a result, sample selection cannot be carried out prior to learning and it has to be done while learning. The absence of sample selection might result in the network over-fitting to the training data.

7.1.2 Meta-Cognitive Sequential Learning Algorithm for Neuro-Fuzzy Inference System (McFIS)

A meta-cognitive sequential learning algorithm for a neuro-fuzzy inference system, referred to as Meta-Cognitive Neuro-Fuzzy Inference System (McFIS), has been proposed. The McFIS is based on Nelson and Narens model of meta-cognition. Similar to this model of meta-cognition, McFIS consists of two components: a cognitive component which is a four layered neuro-fuzzy inference system realizing Takagi-Sugeno-Kang based fuzzy inference mechanism, and a meta-cognitive component which is a self-regulatory learning mechanism. The neuro-fuzzy inference system learns and stores the knowledge from the data and infers the output based on the data. The meta-cognitive learning mechanism efficiently monitors the knowledge in the neuro-fuzzy inference system and controls the learning in the same by deciding *what-to-learn*, *when-to-learn* and *how-to-learn*, efficiently. These three components are realized in McFIS by employing *sample deletion strategy*, *sample reserve strategy* and *sample learning strategy*, respectively. The meta-cognitive component monitors the knowledge in the cognitive component employing prediction error and novelty in the current sample. McFIS employs root mean squared error as a measure of prediction error and spherical potential as a measure of sample novelty. Based on these measures, the meta-cognitive component update its dynamic model of the neuro-fuzzy inference system. In addition, McFIS fully exploits the localization property of a Gaussian rule to achieve a compact rule base with better performance.

The performance of McFIS was evaluated on four problems from the domain of system identification and prediction. The performance was evaluated on sequential adaptive fuzzy inference system [169], evolving Takagi Sugeno based system [12], online sequential fuzzy extreme learning machine [168], and simplified evolving Takagi Sugeno system [13]. The performance comparison indicates an average error reduction of 50% in all the problems compared against. McFIS achieves this accuracy employing less than 50% samples and significantly lesser rules than other algorithms compared against.

The performance of McFIS was also evaluated on a set of eight benchmark binary as well as multi-category classification problems. The performance comparison for classification problem was conducted on sequential adaptive fuzzy inference system [169] and online sequential fuzzy extreme learning machine [169]. It could be observed that McFIS attains better classification accuracy than all other algorithms.

7.1.3 Meta-Cognitive Sequential Learning Algorithm for Neuro-Fuzzy Inference System based Classifier

It has been shown in literature that algorithms developed for solving regression/ approximation problems might not solve classification problems, efficiently. In order to solve classification problems efficiently, class-specific information is essential. Therefore, McFIS has been modified to solve classification problems (McFIS-classifier), by employing class-specific information. Similar to McFIS, McFIS-classifier consists of a cognitive component which is a four layered neuro-fuzzy inference system and a meta-cognitive component which is a self-regulatory learning mechanism. In McFIS-classifier, the self-regulatory learning mechanism efficiently monitors the class-specific knowledge in the neuro-fuzzy inference system and self-regulates the learning by employing the control strategies of *sample deletion strategy*, *sample reserve strategy* and *sample learning strategy*. These control strategies are decided based on class-specific information.

During monitoring, in order to measure error and novelty, we employ hinge-loss based error function and class-specific spherical potential. It has been shown in literature that for classification problems, hinge-loss function estimates posterior probability more accurately than mean squared error function. McFIS-classifier considers class-specific information during rule addition, rule update as well as rule pruning. During rule addition, the width of the newly assigned rule is initialized by considering its distance to the nearest rule belonging to the same class and to that of other classes. The width is assigned with the aim of minimizing misclassification among rules representing different classes while improving generalization among rules representing same class. The output weight is initialized by considering hinge-loss based error and Gaussian rule exploitation. During rule parameter update, the nearest rule belonging to the same classes is updated employing a decoupled extended Kalman filter. A rule belonging to the same class is pruned from the network, if it consistently under-performs for samples belonging to the same class.

The performance of McFIS-classifier has been evaluated on a set of fourteen benchmark binary (five) as well as multi-category (nine) classification problems. A quantitative performance evaluation with other classifiers including support vector machines [39],

evolving classifier [17], online sequential fuzzy extreme learning machine [168], sequential adaptive fuzzy inference system [169], indicate the superior performance of McFIS-classifier on both binary classification as well as multi-category classification problems, based on overall, geometric mean accuracy and F1-score. A qualitative performance comparison based on one-way repeated measures ANOVA and posthoc Dunnett test reinforce the superiority of McFIS-classifier in learning the underlying decision surface based on examples.

7.1.4 Meta-Cognitive Neuro-Fuzzy Inference System based classifier for Human Action Recognition

In this thesis, the developed McFIS-classifier is employed for efficiently solving the problem of human action recognition. Optical flow based information has been shown to be able to capture the apparent movement in objects, by design. Moreover, they are also shown to be able to represent actions hierarchically from local pixel level to global object level. In this thesis, such an optical flow based feature was extracted to represent human action, efficiently. The decision boundary between different actions based on these optical flow based features are approximated by McFIS-classifier. The performance of such a human action recognition system was evaluated on two benchmark Weizmann and KTH video action sequences. The performance comparison with standard support vector machines based classifier indicates that McFIS-classifier outperforms it by 20% to 30% by employing fewer rules at frame level. In sub-sequence level detection, McFIS-classifier outperforms support vector machines by atleast 12%.

The McFIS based action recognition system has been further extended to detect actions from 3D video sequences. 3D optical flow based features were extracted from 2D optical flow based feature by combining with them with depth flow information from the 3D video. The performance of such a 3D action recognition system was evaluated on a proprietary action recognition data sets with eight subjects performing a set of eight actions. A set of 3241 action sequence was obtained. A 10-fold cross validation was performed with 75% samples for training. The performance was compared against standard support vector machines [39] and extreme learning machine [84]. It was observed that McFIS-classifier outperformed other classifiers by around 2% - 10%.

7.1.5 Meta-Cognitive Neuro-Fuzzy Inference System based classifier for Human Emotion Recognition

A meta-cognitive neuro-fuzzy inference system based classifier has been extended to solve the problem of facial emotion recognition. In literature, various features representation techniques are available such as pixel based representation, local binary patterns, Gabor filter based, scale invariant feature transform, and curvelet based features. Each of these representation techniques come with their own advantages and shortcoming. In this thesis, a comparative study of various feature representation techniques have been conducted to evaluate their performance with McFIS-classifier. Moreover, two variants of database independent study have also been conducted to validate the generalization ability of McFIS and representability of each of the employed features. The performance was studied on two standard data sets widely used, Japanese female facial expression data base and Taiwanese facial expression identification data base. The performance was compared against standard support vector machine classifier. The performance comparison indicates the superior classification and generalization ability of McFIS.

7.2 Future Directions

The studies conducted in this thesis could be extended in various respects. A few of these possible research directions are explained below under various heads.

7.2.1 Improvements from Meta-Cognitive Principles

In this thesis, the model of meta-cognition employed has been adapted from the model proposed by Nelson and Narens [145]. In their model, human meta-cognitive process is divided into cognitive and meta-cognitive components. Between these components there are two types of signals, monitory signal and control signal. These components and signals together influence entire process of learning, starting from knowledge acquisition to retention and finally, knowledge retrieval. The framework of human meta-memory as per Nelson and Narens is provided in Fig. 7.1. In this thesis, only the control strategies of *selection of kind of processing*, *allocation of study time* and *termination of study*, have been modeled. Other aspects of meta-cognition also needs to be considered for developing a human-like meta-memory structure.

Figure 7.1: Theoretical Framework of Monitoring and Control of Human Meta-Memory adapted from [145]

7.2.1.1 Ease-of-Learning Judgements

Ease-of-learning judgements as the name suggests, determines how easy or tough a particular knowledge is to learn. According to educational psychology, this judgement is carried out prior to learning a given task. In neuro-fuzzy inference systems, one of the possible realizations of ease-of-learning could be to calculate the data complexity, what kind of learning to employ. In addition, these process could be realized as selection of the network parameters, allocation of resources, and processing time required for learning.

7.2.1.2 Feeling-of-Knowing Judgements

Feeling-of-knowing judgement occurs during learning process. The feeling of knowing refers to predictions about subsequent memory performance on previously non-recalled items. These judgements could be viewed as a measure of the amount of learning required. In the existing McFIS, these feeling-of-knowing judgements can help in selecting or adapting the add, delete, parameter update as well as novelty threshold. One of the initial works with such a realization of feeling-of-knowing judgements is available in [199].

7.2.1.3 Judgement of Knowing

In humans, judgement of knowing refers to the one's estimate of how well they have learnt an item. It is the introspection involved about one's learning. In a neuro-fuzzy

framework, these judgements can be employed for evaluating the knowledge acquired by the system and more importantly, it can evaluate if the newly gained knowledge has destroyed any existing knowledge. In McFIS, one can employ the reserve samples or deleted samples for making such a judgement. In McFIS, deleted samples are not being efficiently utilized in any aspect of learning.

7.2.1.4 Modeling Knowledge Retention

In literature of evolving systems, it has been shown that one of the many problems faced by evolving and adaptive systems is that of drifts and shifts in data. In humans, the knowledge gained will be forgotten over time, if it is not recalled often. However, once learnt, a knowledge is easily recollective. Similarly, in a neuro-fuzzy inference system, the knowledge need to be learnt and with time it should be forgotten. However, this forgotten memory should be recollective. Moreover, the process of forgetting should be gradual and like in human beings should be dependent on factors like strength of original information, its recency, frequency and age.

7.2.1.5 Modeling Knowledge Retrieval

It involves recollection of all the relevant knowledge to the given input. According to Nelson and Narens, it involves a self-directed search with the help of a selection strategy and termination of search upon successful or unsuccessful recollection. These processes are guided by the monitory signal of confidence in retrieved answers. One needs to realize and model these aspects in a soft computing framework. In literature of psychology, various other models of meta-cognition has been proposed which deals with different aspects of cognition, such as collaboration among different individuals, deciding on operational strategy, etc. [40]. A detailed analysis of these models of meta-cognition needs to be conducted and computational models for them need to be found.

7.2.2 Improvements in Fuzzy Inference Mechanism

7.2.2.1 Application Specific Inference Mechanism

In this thesis, a neuro-fuzzy inference system employing Takagi-Sugeno-Kang fuzzy inference mechanism is employed. In literature various other inference mechanisms have also

been proposed which cater to different group of problems, such as Mamdani inference mechanism for rule interpretability. Moreover, efficient inference mechanisms need to be found which can cater to the twin requirement of accuracy and interpretability. In a meta-cognitive learning framework, the learning algorithm should be able to choose the type of inference mechanism to be employed based on the data and performance requirements.

7.2.2.2 Curse of Dimensionality

A pressing issue in existing fuzzy inference mechanisms is the curse of dimensionality in features. As the input feature dimension increases, the total membership of a particular rule decreases, implying the confidence of a sample belonging to a given rule decreases. Let us consider a sample with input dimensionality of 90. Let us also assume that for a given rule, each input feature has a membership of 0.95. However the total membership of the rule, in a Takagi-Sugeno-Kang framework, is less than 0.01 (0.95^{90}). Also, if one of the input features spuriously get a membership close to zero, the total contribution of the rule will tend to zero. This phenomenon may adversely affect the performance of the network. Hence, there is a need to design fuzzy inference mechanism which is able to handle large number of features and be able to handle certain noise in the data without degrading the performance significantly.

7.2.2.3 Membership Functions

In this thesis, a neuro-fuzzy inference system employing Gaussian membership function is employed. In literature various other probable membership functions have been suggested, such as elliptical basis function, inverse quadratic basis function, Cauchy basis function, etc. Each of these membership functions have their own advantages. These membership functions should be considered problem-wise. Recently, a q -Gaussian membership function has been proposed [55] which can realize various radial basis function network by varying its q parameter. The effect of using such a membership function, with the learning algorithm deciding on the type of membership to be realized, should be studied.

7.2.2.4 Fuzzy Sets

Currently, McFIS employs a Type-1 fuzzy set, wherein the memberships are crisp. However, in many real-world problems there are various noise in the data and the data itself is non-stationary. In order to handle such data non-stationarity, Type-2 fuzzy set theory has been proposed in literature. This fuzzy set approach handles the non-stationarity in the input signals employing a secondary membership function for a given primary membership function. These concepts have been extended to the domain of neuro-fuzzy inference systems [96, 202]. However, the recent works on Type-2 neuro-fuzzy inference system lacks automatic selection of the type of membership realization (membership modeling uncertain mean or modeling uncertain standard deviation), selection of type-reduction approaches. In addition, only preliminary works on introducing the concepts of meta-cognition in modeling Type-2 neuro-fuzzy systems are available [202].

7.2.3 Improvements in Neural Learning Algorithms

7.2.3.1 Computational Complexity

McFIS proposed in this thesis employs a sequential learning algorithm with extended Kalman filter based parameter update criteria, in a supervised learning framework. The use of meta-cognitive learning principles has helped it in reducing the training time, significantly. However, in presence of tall or fat data with large number of samples or large number of features and classes, McFIS may fail or become slower, due to the use of extended Kalman filter based approach to adapt the learning parameters. The main bottle neck in employing Kalman filtering approach is the size of error covariance matrix which is directly proportional to the number of rules and number of features. In the learning aspects, there is a need to find an efficient substitute of computationally costly extended Kalman filter based technique.

In literature a mean squared error minimization based technique has been employed in literature, known as projection based learning. It solves a set of linear equations, making the computation cheaper. Such a projection based learning algorithm has been employed for initializing and adapting the rules of the neuro-fuzzy inference system [199]. However, projection based learning algorithm renders the learning offline, as it requires

the history of entire samples for learning. There is a need to extend the projection based learning algorithm to handle sequential data.

7.2.3.2 Cost Functions

The proposed McFIS and McFIS-classifier employs an error based cost function and a hinge-loss error based cost function, respectively. However, in literature, various other cost functions have been proposed with various capabilities. Two such cost functions include entropy based cost function and logarithmic error based cost function. Entropy based cost function has been shown to constrain all the moments of error unlike mean squared error, which only minimizes based on second moment of error. Such a neuro-fuzzy learning algorithm employing error entropy minimization has been proposed in [203]. These and similar cost function needs to be studied and need to be included in McFIS, in order to improve its performance.

7.2.4 Issues Related to Input and Output Features

7.2.4.1 Use of Complex-Valued Signals

Neural networks and fuzzy inference systems are universal approximators whose applications have been investigated widely. However, for many practical problems, mathematical system identification is difficult due to complexity and non-linearity. Converting real-valued problems to complex-valued ones and solving them in complex domain is an emerging area of research, due to the advantage in the non-linear and higher dimensional mapping when signals are converted from real-valued domain to complex-valued domain. Research in the field of complex fuzzy sets [163, 46] are being conducted. Complex fuzzy sets are an extension of fuzzy sets, wherein the membership of each element of a complex fuzzy set is extended to complex-valued state. In a complex fuzzy set, membership values are complex numbers in the unit disc of the complex plane. Although theory of complex fuzzy sets are available, only few works are available in developing a complex-valued neuro-fuzzy inference system is available in literature [230, 200, 201]. One of the future research directions could be extending the concepts of meta-cognitive learning to the domain of complex-valued neuro-fuzzy inference systems.

7.2.4.2 Efficient Feature Selection

In this thesis, a meta-cognitive neuro-fuzzy inference system is proposed which combines sample selection and model selection, for building an efficient and parsimonious neuro-fuzzy inference system. Another important aspect in building an efficient neuro-fuzzy inference system is feature selection. In a sequential learning framework, it is assumed that the data for training the network is not available *a priori*. As a result, feature selection from the data is not possible before training. Thus there is a need to build a learning algorithm which is able to perform sample selection, feature selection and model selection, for building an efficient neuro-fuzzy inference system. A meta-cognitive learning framework realizing this aspect needs to be built.

7.2.4.3 Ability to Handle Varying Inputs and Outputs

In an evolving system, the input features and the output features might vary over time. An adaptive, evolving learning algorithm must be able to handle such a variation in the input and the output features.

Author's Publications

Journals

1. K. Subramanian and S. Suresh, "A Meta-Cognitive Sequential Learning Algorithm for Neuro-Fuzzy Inference System", *Applied Soft Computing*, 12(11), 3603 - 3614, 2012.
2. K. Subramanian and S. Suresh, "Human Action Recognition Using Meta-Cognitive Neuro-Fuzzy Inference System", *International Journal of Neural System*, 22(6), 1250028(15), 2012.
3. K. Subramanian, S. Suresh and N. Sundararajan, "A Meta-Cognitive Neuro-Fuzzy Inference System (McFIS) for Sequential Classification Problems", *IEEE Transactions in Fuzzy System*, 21(6), 1080 - 1095, 2013.
4. K. Subramanian, R. Savitha and S. Suresh, "A Complex-Valued Neuro-Fuzzy Inference System and Its Learning Mechanism", *Neurocomputing*, 123, 110 - 120, 2014.
5. K. Subramanian, A. K. Das, S. Suresh and R. Savitha, "A Meta-Cognitive Interval Type-2 Fuzzy Inference System and its Projection Based Learning Algorithm", *Evolving Systems*, DOI:10.1007/s12530-013-9102-9, 2014.
6. K. Subramanian, R. Savitha, S. Suresh, "A Meta-Cognitive Complex-Valued Interval Type-2 Fuzzy Inference System" *IEEE Transactions on Neural Networks and Learning Systems*, **Accepted**

Conference Proceedings

1. S. Suresh and K. Subramanian, “A sequential learning algorithm for Meta-Cognitive Neuro-Fuzzy Inference System for classification problems”, *International Joint Conference on Neural Networks*, 2507 - 2512, San Jose, California, 2011.
2. K. Subramanian and S. Suresh, “A Meta-Cognitive Neuro-Fuzzy Inference System for Human Emotion Recognition”, *International Joint Conference on Neural Networks*, 1 - 7, Brisbane, Australia, 2012.
3. K. Subramanian and S. Suresh, “Human Action Recognition using Meta-Cognitive Neuro-Fuzzy Inference System”, *International Joint Conference on Neural Networks*, 1 - 8, Brisbane, Australia, 2012.
4. K. Subramanian, R. Savitha and S. Suresh, “A Complex Neuro-Fuzzy Inference System for Wind Prediction”, *International Joint Conference on Neural Networks*, 1 - 7, Brisbane, Australia, 2012.
5. K. Subramanian, R. Savitha, S. Suresh and B. S. Mahanand, “Complex-Valued Neuro-Fuzzy Inference System based Classifier”, *International Conference on Swarm, Evolutionary and Memetic Computing*, Lecture Notes in Computer Science (7677), 348 - 355, India, 2012.
6. K. Subramanian, R. Savitha and S. Suresh, “A Meta-Cognitive Interval Type-2 Fuzzy Inference System Classifier and its Projection Based Learning Algorithm”, *IEEE Conference on Evolving and Adaptive Intelligent Systems*, 48 – 55, Singapore, 2013.
7. K. Subramanian, R. Savitha and S. Suresh, “Zero-Error Density Maximization based Learning Algorithm for a Neuro-Fuzzy Inference System”, *IEEE International Conference on Fuzzy Systems*, 1 - 7, Hyderabad, India, 2013.
8. K. Subramanian and S. Suresh, “A Projection Based Learning Algorithm for Meta-Cognitive Neuro-Fuzzy Inference System”, *IEEE International Conference on Fuzzy Systems*, 1 - 8, Hyderabad, India, 2013.

9. K. Subramanian, R. V. Babu and S. Suresh, “An Optical Flow Feature and Mc-FIS Based Approach for 3-Dimensional Human Action Recognition”, *International Symposium Series on Neural Information Processing*, 1 - 6, Singapore, 2014.
10. K. Subramanian, R. V. Babu, R. Savitha, “Database Independent Human Emotion Recognition with Meta-Cognitive Neuro-Fuzzy Inference System”, *International Symposium Series on Neural Information Processing*, 1 - 6, Singapore, 2014.

Bibliography

- [1] S. Abbas, M. S. Khan, K. Ahmed, M. Abdullah, and U. Farooq. Bio-inspired neuro-fuzzy based dynamic route selection to avoid traffic congestion. *International Journal of Scientific and Engg. Research*, 2(6):1 – 6, 2011.
- [2] H. Adeli and S.L. Hung. An adaptive conjugate gradient learning algorithm for effective training of multilayer neural networks. *Applied Mathematics and Computation*, 62(1):81 – 102, 1994.
- [3] H. Adeli and X. Jiang. Dynamic fuzzy wavelet neural network model for structural system identification. *Journal of Structural Engineering*, 132(1):102 – 111, 2006.
- [4] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037 – 2041, 2006.
- [5] A. Ajith and N. Baikunth. Evolutionary design of neuro-fuzzy systems - a generic framework. In *Japan- Australia joint workshop on intelligent and evolutionary systems*, pages 106–113, 2000.
- [6] J.S. Albus. New approach to manipulator control: The cerebellar model articulation controller. *ASME Journal of Dynamic Systems, Measurement, and Control*, 97(3):220 – 227, 1975.
- [7] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):288 – 303, 2010.

- [8] M. Alizadeh, R. Rada, F. Jolai, and E. Fotoohi. An adaptive neuro-fuzzy system for stock portfolio analysis. *International Journal of Intelligent Systems*, 26(2): 99–114, 2011.
- [9] J. Andreu, R. D. Baruah, and P. Angelov. Real time recognition of human activities from wearable sensors by evolving classifiers. In *IEEE International Conference on Fuzzy Systems*, pages 2786–2793. IEEE, 2011.
- [10] P. Angelov. *Evolving Takagi-Sugeno fuzzy systems from streaming data (eTS+)*. New York: John Wiley & Sons, 2010.
- [11] P. Angelov. *Autonomous Learning Systems*, chapter Autonomous Learning Classifiers, pages 133 – 141. John Wiley & Sons, Ltd, 2012.
- [12] P. Angelov and D. Filev. An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics*, 34(1):484 – 498, 2004.
- [13] P. Angelov and D. Filev. SIMPL_eTS: A simplified method for learning evolving Takagi-Sugeno fuzzy models. In *IEEE International Conference on Fuzzy Systems*, pages 1068 – 1073, 2005.
- [14] P. Angelov and R. Yager. A new type of simplified fuzzy rule-based system. *International Journal of General Systems*, 41(2):163 – 185, 2012.
- [15] P. Angelov and X. Zhou. Evolving fuzzy systems from data stream in real time. In *International Symposium on Evolving Fuzzy Systems*, pages 29 – 35, 2006.
- [16] P. Angelov and X. Zhou. Evolving fuzzy rule based classifiers from data streams. *IEEE Transactions on Fuzzy Systems*, 16(6):1462–1475, 2008.
- [17] P. Angelov, E. Lughofer, and X. Zhou. Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*, 159(23):3160 – 3182, 2008.
- [18] M. Arvaneh, C. Guan, K. K. Ang, and C. Quek. Robust EEG channel selection across sessions in brain-computer interface involving stroke patients. In *International Joint Conference Neural Networks*, pages 1 – 6, 2012.

- [19] M. M. Asif, P. Angelov, and H. Ahmed. An approach to real-time color-based object tracking. In *International Symposium on Evolving Fuzzy Systems*, pages 86–91. IEEE, 2006.
- [20] R.V. Babu, B. Anantharaman, K.R. Ramakrishnan, and S.H. Srinivasan. Compressed domain action classification using HMM. *Pattern Recognition Letters*, 23(10):1203 – 1213, 2002.
- [21] R.V. Babu, S. Suresh, and R. Savitha. Human action recognition using a fast learning fully complex-valued classifier. *Neurocomputing*, 89:202 – 212, 2012.
- [22] Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.
- [23] Al. N. Bassam, M. A. Mallouh, and A. K. Abed. Automated intelligent diagnostic of alzheimer disease based on neuro-fuzzy system and discrete wavelet transform. *Biomedical Engineering: Applications, Basis and Communications*, 0(0):1450035, 0. doi: 10.4015/S1016237214500355.
- [24] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509 – 522, 2002.
- [25] P. Bilinski and F. Bremond. Contextual statistics of space-time ordered features for human action recognition. In *IEEE International Conference Advanced Video and Signal-Based Surveillance*, 2012.
- [26] C. Blake and C. Merz. UCI repository of machine learning databases, 1998. URL <http://archive.ics.uci.edu/ml/>. Department of Information and Computer Sciences, University of California, Irvine.
- [27] A.F. Bobick and J.W Davis. The recognition of human movement using temporal templates. *IEEE Transactions on on Pattern Analysis and Machine Intelligence*, 23(3):257 – 267, 2001.
- [28] Y. Bodyanskiy and O. Vynokurova. Hybrid adaptive wavelet-neuro-fuzzy system for chotic time seried identification. *Information Sciences*, 220(1):170 – 179, 2013.

- [29] J. Borkowski, M. Carr, and M. Pressely. “Spontaneous” strategy use: Prespectives from metacognitive theory. *Intelligence*, 11(1):61 – 75, 1987.
- [30] I. Buciu, C. Kotropoulos, and I. Pitas. ICA and Gabor representation for facial expression recognition. In *IEEE International Conference on Image Processing*, volume 2, pages 855 – 858, 2003.
- [31] Martin D Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003.
- [32] A. J. Calder, A. M. Burton, P. Miller, A. W. Young, and S. Akamatsu. A principal component analysis of facial expression. *Vision Research*, 41:1179 – 1208, 2001.
- [33] G.A. Carpenter, S. Grossberg, and J.H. Reynolds. ARTMAP: supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4(1):565 – 588, 1991.
- [34] G. Castellano, A.M. Fanelli, and C. Mencar. An empirical risk functional to improve learning in a neuro-fuzzy classifier. *IEEE Transactions on Systems, Man, Cybernetics, Part B: Cybernetics*, 34(1):725 – 731, 2004.
- [35] Z.S.H. Chan and N. Kasabov. Evolutionary computation for online and offline parameter tuning of evolving fuzzy neural networks. *International Journal of Computational Intelligence and Applications*, 4(3):309 – 319, 2004.
- [36] L.-F. Chen and Y.-S. Yen. Taiwanese facial expression image database. Brain Mapping Laboratory, Institute of Brain Science, National Yang-Ming University, Taipei, Taiwan, 2007.
- [37] X. Chen, D. Li, Y. Bai, and Z. Xu. Modeling and neuro-fuzzy adaptive attitude control for eight rotor MAV. *International Journal on Control, Automation and Systems*, 9(6):1154 – 1163, 2011.
- [38] V. Cirovic and D. Aleksendric. Adaptive neuro-fuzzy wheel slip control. *Expert Systems with Applications*, 40(13):5197 – 5209, 2013.

- [39] C Cortes and V Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [40] M.-T. Cox. meta-cognition in computation: A selected reserch review. *Artificial Intelligence*, 169(2):104 – 141, 2005.
- [41] K. Cpalka, O. Rebrova, R. Nowicki, and L. Rutkowski. On design of flexible neuro-fuzzy systems for non-linear modelling. *International Journal on General Systems*, 42(6):706 – 720, 2013.
- [42] R.-S. Croder. Prediction the Mackey-Glass time series with cascase correlation learning. In *Connectionist Models Summer School*, pages 117 – 123. Carnegic Mellon University, 1990.
- [43] E. Czogala and J. Leski. *Studies in Fuzziness and Soft Computing*, chapter Fuzzy and Neuro-Fuzzy Intelligent Systems. Psysica-Verlag, Heidelberg, 2000.
- [44] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 886 – 893, 2005.
- [45] S. Das and M.K. Kundu. A neuro-fuzzy approach for medical image fusion. *IEEE Transactions on Biomedical Engineering*, 60(12):3347–3353, 2013.
- [46] S. Dick. Toward complex fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 13(3): 405 – 414, 2005.
- [47] G. Donato, M. S. Bartlett, J.C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 21(10):974 – 989, 1999.
- [48] Paul E. and Erika R. L. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, 1997.

- [49] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, volume 2, pages 726 – 733, 2003.
- [50] P. Ekman. Strong evidence for universals in facial expressions: A reply to Russell’s mistaken critique. *Psychological Bulletin*, 115(2):268 – 287, 1994.
- [51] P. Ekman and W. V. Friesen. Constants across cultures in the face and emotions. *Journal of Personality Social Psychology*, 17(2):124 – 129, 1971.
- [52] P. Ekman, W. V. Friesen, and P. Ellsworth. What emotion categories or dimensions can observers judge from facial behavior. In *Emotion in the human face*, P. Ekman (Ed.), pages 39 – 55. Cambridge University Press, 1982.
- [53] I. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757 – 763, 1997.
- [54] P. O. Fanger. *Thermal comfort - Analysis and application in environmental engineering*. New York: McGraw - Hill, 1970.
- [55] F. Fernández-Navarro, C. Hervás-Martínez, P. A. Gutiérrez, J. M. Peña-Barragán, and F. López-Granados. Parameter estimation of q-gaussian radial basis functions neural networks with a hybrid algorithm for binary classification. *Neurocomputing*, 75(1):123–134, 2012.
- [56] J. H. Flavell. Metacognitive aspects of problem-solving. In L. B. Resnick (Ed.), *The nature of intelligence*, pages 231 – 236, 1976.
- [57] J. H. Flavell. *Speculations about the nature and development of metacognition*. In F. E. Weinert & R. H. Kluwe (Eds.). *Metacognition Motivation and Understanding*, 1987.
- [58] J.H. Flavell. Meta-cognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, 34(10):906 – 911, 1979.

- [59] J.H. Flavell. *Cognitive Development*. Prentice Hall: Great Britain, 1988.
- [60] L. Fortuna, G. Rizzotto, M. Lavorgna, G. Nunnari, M. G. Xibilia, and R. Caponetto. Neuro-fuzzy strategies for monitoring urban traffic noise. In *Soft Computing*, Advanced Textbooks in Control and Signal Processing, pages 209–220. 2001.
- [61] T. Furuhashi. *Advances in fuzzy logic, neural networks, and genetic algorithms*. IEEE/Nagoya-University World Wisepersons Workshop. Springer, 1994.
- [62] M. Gao, X. Hong, and C. J. Harris. A neuro-fuzzy classifier for two class problems. In *UK workshop on computational intelligence*, pages 1 – 6. 2012.
- [63] E.-P. George and G.-M. Jenkins. *Time Series Analysis, Forecasting and Control*. CA: Holden Day, San Francisco, 1976.
- [64] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Behavior classification by eigen decomposition of periodic motions. *Pattern Recognition*, 38(7):1033 – 1043, 2005.
- [65] J. Gomez and D. Dasgupta. Evolving fuzzy classifier for intrusion detection. In *IEEE Workshop on Information Assurance*, number 3 in 6, pages 321 – 323, 2002.
- [66] M.A. Gonzalex-Olvera, A.G. Gallardo-Hernandez, Y. Tang, M.C. Revilla-Monsalve, and S. Islas-Andrade. A discrete time recurrent neuro-fuzzy network for black box modeling of insullin dynamics in diabetic type-1 patients. *International Journal of Neural Systems*, 20(2):149 – 158, 2010.
- [67] M. A. O. Gonzalez, A. G. Gallardo, Y. Tang, M. C. Revilla, and S. Islas. A discrete time recurrent neuro-fuzzy network for black box modeling of insulin dynamics in diabetic type-1 patients. *International Journal of Neural Systems*, 20(2):149 – 158, 2010.
- [68] L. Gorelick, M. Galun, E. Sharon, A. Brandt, and R. Basri. Shape representation and classification using poisson equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1991 – 2005, 2006.

- [69] K. Guo, P. Ishwar, and J. Konrad. Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels. In *Recognizing Patterns in Signals, Speech, Images and Videos*, pages 294 – 305. Springer Berlin, 2010.
- [70] A. Hadid, M. Pietikainen, and T. Ahonen. A discriminative feature space for detecting and recognizing faces. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 797 – 804, 2004.
- [71] A.M.A Haidar, A. Mohamed, M. Al-Dahhagh, A. Aini Hussain, and M. Masoum. An intelligent load shedding scheme using neural networks and neuro-fuzzy. *International Journal of Neural Systems*, 19(6):473 – 479, 2009.
- [72] H. Han and J. Qiao. A self-organizing fuzzy neural network based on a growing and pruning algorithm. *IEEE Transactions on Fuzzy Systems*, 18(6):1129 – 1143, 2010.
- [73] I. Hayash and M. Umamo. Perspectives and trends of fuzzy-neural networks. *Journal of Japan Society for Fuzzy Theory and Systems*, 5(2):178 – 190, 1993. in Japanese.
- [74] I. Hayashi, E. Naito, and N. Wakami. A proposal of fuzzy connective with learning function and its application to fuzzy information retrieval. In *International Fuzzy Engineering Symposium*, pages 446 – 455, 1991.
- [75] K. Hayashi, Y. Shimizu, Y. Dote, A. Takayama, and A. Hirako. Neuro-fuzzy transmission control for automobile with variable loads. *IEEE Transactions on Control Systems Technology*, 3(1):49 – 53, 1995.
- [76] D.O. Hebb. *The Organization of Behavior*. NY: John Wiley & Sons, 1949.
- [77] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the theory of neural computation*. Addison-Wesley: Redwood City, California, 1991.
- [78] W.L. Ho, W. L. Tung, and C. Quek. An evolving Mamdani-Takagi-Sugeno based neural-fuzzy inference system with improved interpretability-accuracy. In *IEEE International Conference Fuzzy Systems*, pages 1 – 8, 2010.

- [79] H. Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.
- [80] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences of the USA*, 1982.
- [81] M. S. Hosseini and M. Zekri. A review of medical image classification using Adaptive Neuro-Fuzzy Inference System (ANFIS). *Journal of Medical Signals and Sensors*, 2(1):49 – 60, 2012.
- [82] C.-C. Hou. A fuzzy cerebellar model articulation controller. In *IEEE International Conference on Fuzzy Systems*, pages 1171 – 1178, 1992.
- [83] M.K. Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8(2):179 – 187, 1962.
- [84] G.B. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on System, Man and Cybernetics, Part B: Cybernetics*, PP(99):1 – 17, 2011.
- [85] H. F. Huang and S. C. Tai. Facial expression recognition using new feature extraction algorithm. *Electronic Letters on Computer Vision and Image Analysis*, 11(1): 41 – 54, 2012.
- [86] K.J. Hunt, D. Sbarbaro, R. Zbikowski, and P.J. Gawthrop. Neural networks for control systems – a survey. *Automatica*, 28(6):1083 – 1112, 1992.
- [87] H. Ichihashi, T. Shirai, K. Nagasaka, and T. Miyoshi. Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method of incremental learning. *Fuzzy Sets and Systems*, 81(8): 157 – 167, 1996.
- [88] J.A Iglesias, P. Angelov, A. Ledezma, and A. Sanchis. Human activity recognition based on evolving fuzzy systems. *International Journal of Neural Systems*, 20(5): 355 – 364, 2010.

- [89] R.L Iman and J.M Davenport. Approximations of the critical region of the friedman static. *Communication in Statistics*, 9(6):571–595, 1980.
- [90] R. Isaacson and F. Fujita. Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflections on learning. *Journal of the Scholarship of Teaching and Learning*, 6(1):39–55, 2006.
- [91] H. Ishibuchi, R. Fujioka, and H. Tanaka. Classification of fuzzy vectors by neural networks. *Transactions on of the Institute of Systems, Control and Information Engineering*, 5(5):198 – 206, 1992.
- [92] A. Jalal, Md. Z. Uddin, J. T. Kim, and T.-S. Kim. Daily human activity recognition using depth silhouettes and R transformation for smart home. In *Towards Useful Services for Elderly and People with Disabilities*, volume 6719 of *Lecture Notes in Computer Science*, pages 25 – 32. Springer, 2011.
- [93] J.-S.-R Jang. ANFIS: Adaptive network based fuzzy inference system. *IEEE Transactions on System, Man and Cybernetics*, 23(3):665 – 685, 1993.
- [94] J.S.R. Jang and C.T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks*, 4(1):156–159, 1993.
- [95] X. Jiang and H. Adeli. Dynamic fuzzy wavelet neuroemulator for nonlinear control of irregular building structures. *International Journal for Numerical Methods in Engineering*, 74(7):1045 – 1066, 2008.
- [96] C. F. Juang. A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning. *IEEE Transactions on Fuzzy Systems*, 16(6): 1411 – 1424, 2008.
- [97] C.-F. Juang and C.-T. Lin. An online self-constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, 10(2):144 – 154, 1998.

- [98] C. iF. Juang, S. H. Chiu, and S. W. Chang. A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems. *IEEE Transactions on Fuzzy Systems*, 15(5):998–1008, 2007.
- [99] A. Karim and H. Adeli. Comparison of fuzzy wavelet rbfnf freeway incident detection model with the california algorithm. *Journal of Transportation Engineering*, 128(1):21 – 30, 2002.
- [100] N. Kasabov. Evolving fuzzy neural networks for supervise/unsupervised online knowledge-based learning. *IEEE Transactions on System, Man and Cybernetics, Part B: Cybernetics*, 31(6):902 – 918, 2001.
- [101] N. Kasabov and Q. Song. Dynamic evolving fuzzy neural networks with ‘m-out-of-n’ activation nodes for online adaptive systems. Technical report, Dept. of Information Science, University of Otago, 1999.
- [102] E. Kayacan, Y. Oniz, A.-C. Aras, O. Kaynak, and R. Abiyev. A servo system control with time varying and nonlinear load conditions using type-2 tsf fuzzy neural system. *Applied Soft Computing*, 11(8):5735 – 5744, 2011.
- [103] J. G. Kelly, P. Angelov, M. J. Walsh, H. M. Pollock, M. A. Pitt, P. L. Martin-Hirsch, and F. L. Martin. A self-learning fuzzy classifier with feature selection for intelligent interrogation of mid-ir spectroscopy data from exfoliative cervical cytology using selflearning classifier eclass. *International Journal of Computational Intelligence Research*, 4(4):392–401, 2008.
- [104] J. M. Kelly, P. Angelov, J. Trevisan, A. Vlachopoulou, E. Paraskevaidis, P. L. Martin-Hirsch, and F. L. Martin. Robust classification of low-grade cervical cytology following analysis with ATR-FTIR spectroscopy and subsequent application of self-learning classifier eClass. *Analytical And Bioanalytical Chemistry*, 398(5):2191 – 2201, 2010.
- [105] J. Kim and N. Kasabov. Hyfis: Adaptive neuro-fuzzy inference system and their application to nonlinear dynamical systems. *Neural Networks*, 12(9):1301 – 1319, 1999.

- [106] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim. Support vector machines for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1542 – 1550, 2002.
- [107] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59 – 69, 1982.
- [108] P. Koprinkova-Hriatova. Backpropagation through time training of a neuro-fuzzy controller. *International Journal of Neural Systems*, 20(5):405 – 419, 2010.
- [109] B. Kosko. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24(1):65 – 75, 1986.
- [110] I. Kotsia and I. Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machine. *IEEE Transactions on Image Processing*, 16(1):172 – 187, 2007.
- [111] S. M. Lajevardi and M. Lech. Averaged Gabor filter features for facial expression recognition. In *Digital Image Computing: Techniques and Applications*, pages 71 – 76, 2008.
- [112] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107 – 123, 2005.
- [113] D. Leite, P. Costa, and F. Gomide. Evolving granular neural networks from fuzzy data streams. *Neural Networks*, 38:1 – 16, 2013.
- [114] G. Leng, T.-M. McGinnity, and G. Prasad. An approach for online extraction of fuzzy rules using a self-organizing fuzzy neural network. *Fuzzy Sets and Systems*, 150(2):211 – 243, 2005.
- [115] G. Leng, M.T. McGinnity, and G. Prasad. Design for self-organizing fuzzy neural networks based on genetic algorithms. *IEEE Transactions on Fuzzy Systems*, 14(6):755 – 766, 2006.

- [116] X. Li, Er. M.J., B.S. Lim, J.H. Zhou, O.P. Gan, and L. Rutkowski. Fuzzy regression modeling for tool performance prediction and degradation detection. *International Journal of Neural Systems*, 20(5):405 – 419, 2010.
- [117] R. H. Liang and C. C. Cheng. Short-term load forecasting by a neuro-fuzzy based approach. *International Journal of Electrical Power & Energy Systems*, 24(2):103 – 111, 2002.
- [118] C. T. Lin and C. S. G. Lee. Neural network based fuzzy logic control and decision system. *IEEE Transactions on Computing*, 40(12):1320 – 1336, 1991.
- [119] C.J. Lin and C.T. Chen, C.H.and Lin. Efficient self evolving evolutionary learning for neuro fuzzy inference system. *IEEE Transactions on Fuzzy Systems*, 16(6):1476 – 1490, 2008.
- [120] F. Liu and Er. Meng Joo. Design for self organizing fuzzy neural network with extended Kalman filter. In *IEEE International Conference on Control and Automation*, pages 423 – 427, 2010.
- [121] E. Lopez-Rubio, R.M. Luque-Baena, and E. Dominquez. Foreground detection in video sequences with probabilistic self-organizing maps. *International Journal of Neural Systems*, 21(3):225 – 246, 2011.
- [122] D.G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150 – 1157, 1999.
- [123] W.L. Lu and J.J. Little. Simultaneous tracking and action recognition using PCA – HOG descriptors. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, pages 1 – 6, 2006.
- [124] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of DARPA Image Understanding Workshop*, pages 121 – 130, 1981.
- [125] E. Lughofer. FLEXFIS- a robust incremental learning approach for evolving Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 16(6):1393 – 1410, 2008.

- [126] E. Lughofer and P. Angelov. Handling drifts and shifts in online data streams with evolving fuzzy systems. *Applied Soft Computing*, 11(2):2057 – 2068, 2011.
- [127] A. R. Luria. *The working brain: An introduction to neuropsychology*. Basic Books, 1976.
- [128] M.C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287 – 289, 1977.
- [129] E. H. Mamdani. Twenty years of fuzzy control: Experiences gained and lessons learnt. In *Fuzzy Logic Technology and Applications*, IEEE Technology Update Series, pages 19 – 24. R. J. Marks II, Ed., 1994.
- [130] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Human-computer Studies / International Journal of Man-machine Studies*, 7(1):1–13, 1975.
- [131] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. In *Proceedings of the Institution of Electrical Engineers*, volume 121, pages 1585 – 1588, 1974.
- [132] T. Mandal, Q. M. Jonahan Wu, and Y. Yuan. Curvelet based face recognition via dimension reduction. *Signal Processing*, 89(12):2345 – 2353, 2009.
- [133] H. Matti, H. Riitta, I.J. Risto, K. Jukka, and L.V. Olli. Magnetoencephalography theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics*, 65(2):413, 1993.
- [134] M. Meulders, P. D. Boeck, I. V. Mechelen, and A. Gelmen. Probabilistic feature analysis of facial perception of emotions. *Applied Statistics*, 54(4):781 – 793, 2005.
- [135] J. L. Michael, A. Shigeru, M. Kamachi, and J. Gyoba. Coding facial expressions with Gabor wavelets. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200 – 205, 1998.
- [136] M. Minsky and S. Papert. *Perceptrons: An introduction to computational geometry*. Cambridge, MA: MIT press, 1969.

- [137] S. Mitra and Y. Hayashi. NeuroFuzzy rule generation: Survey soft computing framework. *IEEE Transactions on Neural Network*, 11(3):748–768, 2000.
- [138] J. Molina, J. Gara, J. Diego, and J. T. Portillo. Neuro-fuzzy techniques for image tracking. In *Artificial Neural Nets Problem Solving Methods*, volume 2687 of *Lecture Notes in Computer Science*, pages 504–511. 2003.
- [139] J. Moody and C. Darken. Learning with localized receptive fields. In *Connectionist Models Summer School*, pages 133 – 143, 1998.
- [140] J. Moody and C.J. Draken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281 – 294, 1989.
- [141] J. Nathalie and S. Mohak. *Evaluating learning algorithms: A classification perspective*. Cambridge, 2011.
- [142] D. Nauck and R. Kruse. Neuro-fuzzy systems for function approximation. *Fuzzy Sets and Systems*, 101(2):261 – 271, 1999.
- [143] D. Nauck, F. Klawonn, and R. Kruse. *Foundations of Neuro-Fuzzy Systems*. John Wiley & Sons, Inc., New York, NY, USA, 1997.
- [144] T. D. Ndousse. Fuzzy neural control of voice cells in ATM networks. *IEEE Journal on Selected Areas in Communications*, 12(6):1488 – 1494, 1994.
- [145] T.O. Nelson and L. Narens. Metamemory: A theoretical framework and new findings. *Psychology of Learning and Motivation*, 26(PP):125 – 173, 1990.
- [146] K. C. Ng and M. M. Trivedi. A neuro-fuzzy controller for mobile robot navigation and multirobot convoying. *IEEE Transactions on System, Man and Cybernetics Part B: Cybern*, 28(6):829 – 840, 1998.
- [147] B. A. Nowak and R. K. Nowicki. Learning in rough-neuro-fuzzy system for data with missing values. *LNCS: Artificial Intelligence and Lecture Notes in Bioinformatics*, 7203(1):501 – 510, 2012.

- [148] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *IEEE International Conference on Pattern Recognition*, volume 1, pages 582 – 585, 1994.
- [149] S. N. Omkar, S. Suresh, T. R. Raghavendra, and V. Mani. Acoustic emission signal classification using fuzzy c-means clustering. In *IEEE International Conference on Neural Information Processing*, volume 4, pages 1872–1831, 2002.
- [150] J. Ou, X. B. Bai, Y. Pei, L. Ma, and W. Liu. Automatic facial expression recognition using Gabor filter and expression analysis. In *IEEE International Conference on Computer Modeling and Simulation*, pages 215 – 218, 2010.
- [151] K. Padmakumari, K. P. Mohandas, and S. Thiruvengadam. Long term distribution demand forecasting using neuro fuzzy computations. *International Journal of Electrical Power & Energy Systems*, 21(5):315 – 322, 1999.
- [152] K.N. Pantazopoulos, L.H. Tsoukalas, N.G. Bourbakis, M.J. Brun, and E.N. Houstis. Financial prediction and trading strategies using neurofuzzy approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(4):520–531, 1998.
- [153] W. G. Parrot. *Emotions in social psychology*. Psychology Press, Philadelphia, 2000.
- [154] D. Partouche, M. Pasquier, and A. Spalanzani. Intelligent speed adaptation using a self-organizing neuro-fuzzy controller. In *IEEE Intelligent Vehicles Symposium*, pages 846 – 851, 2007.
- [155] L.I. Perlovsky. Neural mechanisms of the mind, aristotle, zadeh and fMRI. *IEEE Transactions on Neural Networks*, 21(5):718 – 733, 2010.
- [156] D. Petkovic, Z. Cojbasic, and V. Nikolic. Adaptive neuro-fuzzy approach for wind turbine power coefficient estimation. *Renewable and Sustainable Energy Reviews*, 28:191 – 195, 2013.

- [157] W. Po-ngae, R. Choomuang, and J. Bhuripanyo. Neuro fuzzy in six dof tele-robotic control. In *IEEE International conference on mechatronics and automation*, pages 959 – 964, 2008.
- [158] R. Polana and R.C. Nelson. Detection and recognition of periodic nonrigid motion. *International Journal of Computer Vision*, 23(3):261 – 282, 1997.
- [159] R. Poppe. A survey on vision based human action recognition. *International J. of Computer Vision*, 28(2/3):976 – 990, 2010.
- [160] A. Qadir, W. Semke, and J. Neubert. Vision based neuro-fuzzy controller for a two axes gimbal system with small UAV. *Journal of Intelligent and Robotic Systems*, 2013. DOI: 10.1007/s10846-013-9865-z.
- [161] C. Quek, K. C. Yow, P. Y. K Cheng, and C. C. Tan. Investment portfolio balancing: application of a generic self-organizing fuzzy neural network (GenSoFNN). *International Systems in Accounting, Finance and Management*, 16(1 – 2):147 – 164, 2009.
- [162] M. Radeerom, C. Srisaan, and M. L. K. Kasemsan. Prediction method for real thai stock index based on neuro-fuzzy approach. In *Trends in Intelligent Systems and Computer Engineering*, volume 6 of *Lecture Notes in Electrical Engineering*, pages 327–347. 2008.
- [163] D. Ramot, M. Friedman, G. Langholz, and A. Kandel. Complex fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 11(4):450 – 461, 2003.
- [164] D.B. Rashmi and P. Angelov. Online learning and prediction of data streams using dynamically evolving fuzzy approach. In *IEEE International Conference on Fuzzy Systems*, pages 1 – 8, 2013.
- [165] M. I. Rey, M. Galende, M. J. Fuente, and G. I. Sainz-Palmero. Checking orthogonal transformations and genetic algorithms for selection of fuzzy rules based on interpretability-accuracy concepts. *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems*, 20(2):159 – 186, 2012.

- [166] W. Rivers. Autonomy at all costs: An ethnography of meta-cognitive self assessment and self management among experience language learners. *Modern Language Journal*, 85(2):279 – 290, 2001.
- [167] C. P. Rodriguez and G. J. Anders. Energy price forecasting in the ontario competitive power system market. *IEEE Transactions on Power Systems*, 19(1):366 – 374, 2004.
- [168] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran. Online sequential extreme learning machine for function approximation and classification problems. *IEEE Transactions on System, Man and Cybernetics, Part B: Cybernetics*, 39(4):1067 – 1072, 2009.
- [169] H.J. Rong, N. Sundararajan, G.B. Huang, and P. Saratchandran. Sequential adaptive fuzzy inference system (SAFIS) for non-linear system identification and prediction. *Fuzzy Sets and Systems*, 157(9):1260 – 1275, 2006.
- [170] H.J. Rong, N. Sundararajan, P. Saratchandran, and G. B. Huang. Adaptive fuzzy-tolerant controller for aircraft autoland under failures. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1586 – 1603, 2007.
- [171] H.J. Rong, N. Sundararajan, G.B. Huang, and G.S. Zhao. Extended sequential adaptive fuzzy inference system for classification problems. *Evolving Systems*, 2(2): 71 – 82, 2010.
- [172] J. J. Rubio. Modified optimal control with a backpropagation network for robotic arms. *IET Control Theory and Applications*, 6(14):2216 – 2225, 2012.
- [173] J.de.J Rubio. SOFMLS: Online self-organizing fuzzy modified least-squares network. *IEEE Transactions on Fuzzy Systems*, 17(6):1296 – 1309, 2009.
- [174] A. Saha and Q. M. J. Wu. Facial expression recognition using curvelet based local binary patterns. In *IEEE International Conference Acoustics Speech and Signal Processing*, pages 2470 – 2473, 2010.

- [175] A. Samant and H. Adeli. Enhancing neural network incident detection algorithms using wavelets. *Computer-Aided Civil and Infrastructure Engineering*, 16(4):239 – 245, 2001.
- [176] K.P. Sankar and D.D. Majumder. Fuzzy sets and decision making approaches in vowel and speaker recognition. *IEEE Transactions on System, Man and Cybernetics*, 7(8):625 – 629, 1977.
- [177] G. Sateesh Babu and S. Suresh. Meta-cognitive neural network for classification problems in a sequential learning framework. *Neurocomputing*, 81(1):86 – 96, 2012.
- [178] G. Sateesh Babu and S. Suresh. Meta-cognitive neural network for classification problems in a sequential learning framework. *Neurocomputing*, 81(1):86 – 96, 2012.
- [179] R. Savitha, S. Suresh, and N. Sundararajan. A self-regulated learning in a fully complex valued radial basis function neural network. In *IEEE International Joint Conference on Neural Networks*, pages 1 – 8, 2010.
- [180] R. Savitha, S. Suresh, and N. Sundararajan. Metacognitive learning in a fully complex-valued radial basis function neural network. *Neural Computation*, 24(5:1): 297 – 328, 2012.
- [181] T. Sawaragi, K. Shibata, O. Katai, S. Iwai, and K. Tsukada. Integrated cognitive architecture for image understanding using fuzzy clustering and structured neural network. In *IEEE International Conference on Fuzzy Systems*, pages 21 – 28, 1992.
- [182] R. Scherer. Ensembles of the mamdani fuzzy systems. *Studies in Fuzziness and Soft Computing*, 288:51 – 59, 2012.
- [183] B. Scholkopf and A.-J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [184] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human action: A local svm approach. In *IEEE International Conference on Pattern Recognition*, pages 32 – 36, 2004.

- [185] S. Seitz and C. Dyer. View invariant analysis of cyclic motion. *International Journal of Computer Vision*, 25(3):231 – 251, 1997.
- [186] C. Shan, S. Gong, and W. M. Peter. Robust facial expression recognition using local binary patterns. In *IEEE International Conference Image Processing*, pages 370 – 373, 2005.
- [187] L. Shao, D. Wu, and X. Chen. Action recognition using correlogram of body poses and spectral regression. In *Proceedings of IEEE International Conference on Image Processing*, page PP, 2011.
- [188] L. Shen, L. Bai, and M. Fairhurst. Gabor wavelets and general discriminant analysis for face identification and verification. *Image and Vision Computing*, 25:553 – 563, 2007.
- [189] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Machine Learning for Computer Vision*, volume 411 of *Studies in Computational Intelligence*, pages 119 – 135. Springer, 2013.
- [190] K. Siminski. *Man-Machine Interactions*, volume 59, chapter Remark on Membership Functions in Neuro-Fuzzy Systems, pages 291 – 297. Springer Berlin Heidelberg, 2009.
- [191] R. Sindal and S. Tokekar. A neuro-fuzzy call admission control algorithm for voice/data traffic in cdma cellular network. In *IEEE International Advances in Computing Conference*, pages 827 – 832, 2009.
- [192] H.-B. Soleimani, C. Lucas, and B. Araabi. Fast evolving neuro-fuzzy model and its application in online classification and time series prediction. *Pattern Analysis & Applications*, 15(3):279 – 288, 2012.
- [193] Q. Song and N. Kasabov. DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2):144 – 154, 2002.

- [194] H. Soyel and H. Demirel. Facial expression recognition based on discriminative scale invariant feature transform. *Electronics Letters*, 46(5):343 – 345, 2010.
- [195] R.J. Strenberg. What should intelligence tests test? implication for a triarchiac theory of intelligence for intelligence testing. *Educational Researcher*, 13(1):5 – 15, 1984.
- [196] K. Subramanian and S. Suresh. Human action recognition using meta-cognitive neuro-fuzzy inference system. In *IEEE International Joint Conference Neural Networks*, pages 1 – 8, Brisbane, Australia, 2012.
- [197] K. Subramanian and S. Suresh. A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system. *Applied Soft Computing*, 12(11):3603 – 3614, 2012.
- [198] K. Subramanian and S. Suresh. Human action recognition using meta-cognitive neuro-fuzzy inference system. *International Journal of Neural System*, 22(6): 1250028, 2012.
- [199] K. Subramanian and S. Suresh. Projection based learning algorithm for meta-cognitive neuro-fuzzy inference system. In *IEEE International Conference on Fuzzy Systems*, pages 1–8, Hyderabad, India, 2013.
- [200] K. Subramanian, R. Savitha, and S. Suresh. Complex-valued neuro-fuzzy inference system for wind prediction. In *IEEE International Joint Conference on Neural Networks*, pages 1–7, Brisbane, Australia, 2012. IEEE.
- [201] K. Subramanian, R. Savitha, S. Suresh, and BS Mahanand. Complex-valued neuro-fuzzy inference system based classifier. In *Lecture Notes in Computer Science: Swarm, Evolutionary, and Memetic Computing*, volume 7677, pages 348–355. Springer Berlin Heidelberg, 2012.
- [202] K. Subramanian, R. Savitha, and S. Suresh. A meta-cognitive interval type-2 fuzzy inference system classifier and its projection based learning algorithm. In *IEEE Conference on Evolving and Adaptive Intelligent Systems*, Symposium Series on Computational Intelligence, pages 48 – 55, Singapore, 2013.

- [203] K. Subramanian, R. Savitha, and S. Suresh. Zero-error density maximization based learning algorithm for a neuro-fuzzy inference system. In *IEEE International Conference on Fuzzy Systems*, pages 1–7, Hyderabad, India, 2013.
- [204] K. Subramanian, S. Sundaram, and N. Sundararajan. A meta-cognitive neuro-fuzzy inference system mcfis for sequential classification problems. *IEEE Transactions on Fuzzy Systems*, 21(6):1080 – 1095, 2013.
- [205] M. Sugeno. An introductory survey of fuzzy control. *Information Sciences*, 36(1): 59 – 83, 1985.
- [206] S. N. Sulzberger, N. N. Tschicholg-Gurman, and S. J. Vestli. FUN: Optimization of fuzzy rule based on systems using neural networks. In *IEEE Conference on Neural Networks*, pages 312 – 316, 1993.
- [207] Z.L. Sun, K.F. Au, and T.M. Choi. A neuro-fuzzy inference system through integration of fuzzy logic and extreme learning machines. *IEEE Transactions on Systems, Man, Cybernetics, Part B: Cybernetics*, 37(5):1321 – 1331, 2007.
- [208] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgbd images. In *Plan, Activity, and Intent Recognition*, 2011.
- [209] S. Suresh, N. Sundararajan, and P. Saratchandran. Risk sensitive loss functions for sparse multi-category classification problems. *Information Sciences*, 179(21): 2621–2638, 2008.
- [210] S. Suresh, N. Sundararajan, and P. Saratchandran. Sequential multi-category classifier using radial basis function networks. *Neurocomputing*, 71(7-9):1345–1358, 2008.
- [211] S. Suresh, K. Dong, and H.J. Kim. A sequential learning algorithm for self-adaptive resource allocation network classifier. *Neurocomputing*, 73(16-18):3012–3019, 2010.
- [212] R.S. Sutton. *Temporal credit assignment in reinforcement learning*. PhD thesis, University of Massachusetts – Amherst, 1984.

- [213] M. Suwa, N. Sugie, and K. Fujimora. A preliminary note on pattern recognition for human emotional expression. In *International Joint Conference on Pattern Recognition*, pages 408 – 410, 1978.
- [214] H. Takagi. Fusion technology of neural networks and fuzzy systems: A chronicled progression from laboratory to our daily lives. *International Journal of Applied Mathematics and Computer Science*, 10(4):647 – 673, 2000.
- [215] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on System, Man and Cybernetics*, 15(1): 116 – 132, 1985.
- [216] H. Takahashi and H. Minami. Subjective evaluation modeling using fuzzy logic and neural networks. In *IFSA Congress*, pages 520 – 523, Seattle, WA, USA, 1989.
- [217] M. Tang and F. Chen. Facial expression recognition and its application based on curvelet transform and PSO-SVM. *Optik- International Journal for Light and Electron Optics*, 124(22):5401 – 5406, 2013.
- [218] H.S. Terrace and J. Metacalfe. *The missing link in cognition: Origin of self-reflective consciousness*. Oxford University Press, 2005.
- [219] D. Theodoridis, Y. Boutalis, and M. Christodoulou. Dynamical recurrent neuro-fuzzy identification schemes employing switching parameter hopping. *International Journal of Neural Systems*, 22(2):1250004 – 16, 2012.
- [220] C. Thureau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1 – 6, 2008.
- [221] Y. Tian, T. Kanade, and J. Cohn. Evaluation of Gabor wavelet based facial action unit recognition in image sequences of increasing complexity. In *IEEE International Conference Automatic Face and Gesture Recognition*, pages 229 – 234, 2002.
- [222] Y. Tian, T. kanade, and J. Cohn. *Handbook of Face Recognition*, chapter Facial Expression Analysis. Springer, 2005.

- [223] M. Tosun and R. Gunturkun. Anesthetic gas control with neuro-fuzzy system in anesthesia. *Expert systems with Applications*, 37(3):2690 – 2695, 2010.
- [224] Y. Tsukamoto. An approach to fuzzy reasoning method. In M. M. Gupta, R. K. Ragade, and R. R. Yager, editors, *Advances in fuzzy set theory and applications*, pages 137 – 149, 1979.
- [225] E. Tulving. *Elements of episodic memory*. Oxford University Press, 1983.
- [226] W. L. Tung and C. Quek. Financial volatility trading using a self-organising neural-fuzzy semantic network and option straddle-based approach. *Expert System with Applications*, 38(5):4668 – 4688, 2011.
- [227] W.L. Tung and C. Quek. A Mamdani-Takagi-Sugeno based linguistic neuroal-fuzzy inference system for improved interpretability-accuracy representation. In *IEEE Intl Conference Fuzzy Systems*, pages 367 – 372, 2009.
- [228] S. Ushakumari, R. Sankaran, and P. S. C. Nair. Adaptive neuro-fuzzy controller for improved performance of a permanent brushless dc motor. In *International Cong. on Fuzzy Systems*, pages 493 – 496, 2001.
- [229] F. Valdez, P. Melin, and O. Castillo. Particle swarm optimization for designing an optimal fuzzy logic controller of a dc motor. In *North American Fuzzy Information Processing Society*, pages 1 – 6, 2012.
- [230] F.J.M. Velo, I. Baturone, R. Senhadji, and S.S. Solano. Tuning complex fuzzy systems by supervised learning algorithms. In *IEEE International Conference on Fuzzy Systems*, pages 226 – 231, 2003.
- [231] R. H. Wai, Z. Yuan, and R. Muthusamy. Fuzzy neural network inherited sliding mode control for robot manipulator including actuator dynamics. *IEEE T. Neural Networks and Learning Systems*, 24(2):274 – 287, 2013.
- [232] R. J. Wai and Z. W. Yang. Adaptive fuzzy neural network control design via a t-s fuzzy model for a robot manipulator including actuator dynamics. *IEEE Transactions on System, Man and Cybernetics Part B: Cybern*, 38(5):1326 – 1346, 2008.

- [233] J.S. Wang and C.S.G Lee. Self adaptive neuro fuzzy inference system for classification applications. *IEEE Transactions on Fuzzy Systems*, 10(6):790 – 802, 2002.
- [234] N. Wang, Er. Meng Joo, X.Y. Meng, and X. Li. An online self-organizing scheme for parsimonious and accurate fuzzy neural networks. *International Journal of Neural Systems*, 20(5):389 – 403, 2010.
- [235] N. Wang, X. Wang, Y. Tan, P. Shao, and M. Han. *Artificial Intelligence and Lecture Notes in Bioinformatics*, volume 7952 of *Lecture Notes in Computer Science*, chapter An improved learning scheme for extracting T-S fuzzy rules from data samples, pages 53 – 60. 2013.
- [236] Y. Wang and G. Mori. Human action recognition by semilattent topic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1762 – 1774, 2009.
- [237] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224 – 241, 2011.
- [238] H. M. Wellman. Metamemory revisited. *M.T.H Chi ed. Contributions to Human development (trends in memory development research)*, 9, 1983.
- [239] A. Wenden. *Learner strategies for learner autonomy*. Great Britain: Prentice Hall, 1998.
- [240] R. A. Wilson. *Quantum Psychology: how brain software programs you and your world*. New Falcon, 1990.
- [241] S. Wu and Er. Meng Joo. Dynamic fuzzy neural networks - a novel approach to function approximation. *IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics*, 30(2):358 – 364, 2000.
- [242] S. Wu, Er. Joo, Meng Joo, and Y. Gao. A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, 9(4):578 – 594, 2001.

- [243] W. Wu and G. Gao. An application of neuro-fuzzy system in remote sensing image classification. In *International Conference Computer Science and Software Engg.*, pages 1069 – 1072, 2008.
- [244] W. Wu, L. Li, J. Yang, and Y. Liu. A modified gradient based neuro-fuzzy learning algrothim and its convergence. *Information Sciences*, 180(9):1630 – 1642, 2010.
- [245] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20 – 27, 2012.
- [246] W. Xue. Facial expression recognition based on Gabor filter and SVM. *Chinese Journal of Electronics*, 15(4A):809 – 812, 2006.
- [247] C. Xydeas, P. Angelov, S. Y. Chiao, and M. Reoullas. Advances in classification of eeg signals via evolving fuzzy classifiers and dependant multiple hmms. *Computers in biology and medicine*, 36(10):1064–1083, 2006.
- [248] Y. Yacoob and L. S. Davis. Recognizing human facial expression from long image sequences using optical flow. *IEEE Transactions on on Pattern Analysis and Machine Intelligence*, 18(6):636 – 642, 1996.
- [249] J. Yamato, J. Ohya, and K. Ishii. Recognitizing human action in time sequential images using hidden markov model. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 379 – 385, 1992.
- [250] K. Yan, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *International Conference Computer Vision*, volume 1, pages 166 – 173, 2005.
- [251] S. Yang and B. Bhanu. Facial expression recognition using emotion avatar image. In *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, pages 866 – 871, 2011.
- [252] H. Yu, T. Xie, S. Paszczynski, and B. M. Wilamowski. Advantages of radial basis function networks for dynamic system design. *IEEE T. Industrial Electronics*, 58(12):5438 – 5450, 2011.

- [253] W. Yu and X. Li. Automated nonlinear system modeling with multiple fuzzy neural networks and kernel smoothing. *International Journal of Neural Systems*, 20(5): 429 – 435, 2010.
- [254] L.A. Zadeh. Fuzzy sets. *Information and Computation/Information and Control*, 8(3):338 – 353, 1965.
- [255] M.A. Zamora-Izquierdo, R. Toledo-Moreo, M. Valdes-Vela, and D. Gil-Galvan. Neuro-fuzzy based maneuver detection for collision avoidance in road vehicles. In *International Conference on the interplay between natural and artificial computation*, Lecture Notes in Computer Science, pages 429 – 438, 2007.
- [256] J. Zhang, H.S.H. Chung, and W.L. Lo. Chaotic time series prediction using a neuro-fuzzy system with time-delay coordinates. *IEEE Transactions on Knowledge and Data Engineering*, 20(7):956 – 964, 2008.
- [257] T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–85, 2003.
- [258] W. B. Zhang and H. B. Ji. Fuzzy extreme learning machine for classification. *Electronics Letters*, 49(7):448 – 450, 2013.
- [259] J. Zhou, Y. Wang, T. Xu, and L. Liu. A novel facial expression recognition based on the curvelet features. In *Pacific-Rim Symposium on Image and Video Technology*, pages 82 – 87, 2010.