An evolving interval type-2 fuzzy inference system for renewable energy prediction intervals

Nguyen, Trong Trung Anh

2018

Nguyen, T. T. A. (2018). An evolving interval type-2 fuzzy inference system for renewable energy prediction intervals. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/88844>

<https://doi.org/10.32657/10220/45996>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**AN EVOLVING INTERVAL TYPE-2 FUZZY
INFERENCE SYSTEM FOR RENEWABLE ENERGY
PREDICTION INTERVALS**

NGUYEN TRONG TRUNG ANH

**INTERDISCIPLINARY GRADUATE SCHOOL
ENERGY RESEARCH INSTITUTE @ NTU (ERI@N)**

2018

**AN EVOLVING INTERVAL TYPE-2 FUZZY
INFERENCE SYSTEM FOR RENEWABLE ENERGY
PREDICTION INTERVALS**

NGUYEN TRONG TRUNG ANH

**Interdisciplinary Graduate School
Energy Research Institute @ NTU (ERI@N)**

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

2018

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....
Date

.....
Student Name

Abstract

Renewable energy is fast becoming a mainstay in today's energy scenario. Some of the main sources of renewable energy are wind, solar in addition to wave, tidal power, etc. The renewable energy sources introduce a significant amount of uncertainty in the energy grid. The main reason is being the inability to forecast the exact energy that could be generated. Due to various natural and artificial factors, it is difficult to predict the amount of the sources in long-term and even in short-term, i.e., there is a lot of uncertainty involved in using these data to build an artificial prediction system. In this thesis, a solution using Interval Type-2 Neuro-Fuzzy Inference System (IT2FIS) is proposed.

In the literature, IT2FIS has been shown to be capable of handling uncertainty associated with the data. However, there is a challenge in developing an evolving IT2FIS which can learn and evolve the architecture automatically. The first contribution in this thesis is the development of an evolving interval type-2 fuzzy inference system (McIT2FIS-GD) to handle the uncertainty in data. The system models input features as uncertainty by employing interval type-2 sets in rule antecedent. The rules are of Takagi-Sugeno-Kang type and the learning algorithm of the system uses first-order gradient descent approach. The system employs a computationally fast interval type-reduction and is capable of evolving its architecture and parameters based on data. The performance of McIT2FIS-GD has been evaluated on a set of benchmark function approximation problems. The results show that the proposed system is able to generalize the underlying functional relationship between the input and output.

In most of the practical problems, the data are being collected over a period of time in a streaming fashion and the data distribution may change with time. The

training data arrives in sequence and the network has to adjust and adapt the parameters according to the variation of the data. Hence, the proposed IT2FIS has been extended to address these issues by developing an adaptive sequential learning algorithm. The data sample arrives sequentially and the learning algorithm decides on an appropriate strategy for learning that sample. The algorithm employs an extended Kalman filter method to adaptively update the parameters of the network and is denoted as McIT2FIS-EKF. Performance evaluation of McIT2FIS-EKF on system identification and time-series prediction problem has clearly highlighted the advantages of the proposed adaptive sequential learning algorithm.

Although interval type-2 can model uncertainty, the current generation could not quantitatively handle this uncertainty. Hence, we combine prediction intervals proposed in the literature and IT2FIS to quantify the uncertainty in data. Moreover, since the renewable energy data are time-series in nature, we extended the work to recurrent IT2FIS by employing a memory structure in the output layer. Sequential learning algorithm with extended Kalman filter has been developed to learn the antecedent and consequent parameters of the fuzzy neural network.

The proposed systems are used in two real-world renewable energy problems: wind prediction and wave prediction. The wave measurement data were collected from GPS-based directional Waveriders deployed offshore Singapore. The experiments are then conducted on the wave energy characteristics and wind speed forecasting problems. The performance studies have indicated the ability of the system in handling and quantifying uncertainty. An improved model of IT2FIS addressing prediction intervals is also intended to be built and evaluated on long-term forecast.

Acknowledgement

First, I would like to extend my respect and utmost gratitude towards my supervisor, Associate Professor Dr. Suresh Sundaram, Nanyang Technological University for his dedication to this research. From the very beginning, his enthusiastic guidance has helped shape my thesis and directed me on the right path. His constructive criticism on the earlier versions of this study has motivated me to strive towards my full potential.

I also want to appreciate my co-supervisor, Dr. Narasimalu Srikanth for his patience and warm encouragements throughout the time of conducting this research. His guidance in using the sensors for measuring the ocean wave engineering science is truly valuable.

My thesis could not have completed without the immense support from lecturers and lab-mates in Computational Intelligence Laboratory, School of Computer Engineering, Nanyang Technological University who have joined helpful discussions and given me insightful comments. A big thank you also goes to my seniors, Dr. Kartick Subramarian and Dr. Ankit Kumar Das for their precious advice and the willingness to share with me their genuine research experiences.

Last but not least, I am forever grateful for my family. My parents and my sister's unconditional love has re-sparked my hope whenever difficulties discouraged me from continuing this thesis ...

Contents

Abstract	i
Acknowledgement	iii
List of Figures	viii
List of Tables	x
Abbreviations	xi
Symbols	xiii
1 Introduction	1
1.1 Background and Motivation	2
1.2 Issues and Challenges	6
1.3 Research Objectives	7
1.4 Research Contributions	9
1.5 Thesis Organization	11
2 Literature Review	14
2.1 Introduction	15
2.2 Lower Upper Bound Estimation Method	16
2.3 A Review of Hybrid Neuro-Fuzzy Inference System.	18
2.3.1 Type-1 Fuzzy Set	19
2.3.2 Type-2 Fuzzy Set	20
2.3.3 Interval Type-2 Fuzzy Set	20
2.3.4 Classification Based on Types of Fuzzy Inference Mechanism	22

2.3.4.1	Mamdani Fuzzy Inference Mechanism	22
2.3.4.2	Takagi-Sugeno-Kang Fuzzy Inference Mechanism	24
2.3.5	Classification Based on Types of Learning Mode	26
2.3.5.1	Batch/Offline Learning	26
2.3.5.2	Sequential/Online Learning	28
2.3.6	Summary	30
3	Meta-cognitive Interval Type-2 Fuzzy Inference System and Its Gradient Descent Learning Algorithm	33
3.1	Introduction	34
3.2	Meta-cognitive Interval Type-2 Neuro-Fuzzy Inference System Gradient Descent	36
3.2.1	Problem Definition	36
3.2.2	Meta-cognitive Interval Type-2 Neuro-Fuzzy Inference System Gradient Descent Architecture	37
3.2.3	Meta-cognitive Gradient Descent Algorithm for McIT2FIS	40
3.2.3.1	Sample Delete Strategy	42
3.2.3.2	Sample Learn Strategy	44
3.2.3.3	Sample Reserve Strategy	51
3.3	Performance Evaluation	52
3.3.1	Performance Measures	52
3.3.2	Identification of a Nonlinear System 1	53
3.3.3	Identification of a Nonlinear System 2	54
3.3.4	Mackey-Glass Time Series Problem	56
3.3.5	Wind Speed Prediction Problem	57
3.4	Summary	60
4	A Sequential Meta-cognitive Learning Algorithm for Interval Type-2 Fuzzy Inference System	61
4.1	Introduction	62
4.2	Problem Definition	63
4.3	Sequential Meta-cognitive Interval Type-2 Fuzzy Inference System Architecture	64
4.4	Meta-cognitive Learning Algorithm based Extended Kalman Filtering	68
4.4.1	Sample Delete Strategy	69
4.4.2	Sample Learn Strategy	69
4.4.2.1	The Rule Adding Criterion	69

4.4.2.2	The Rule Updating Criterion	71
4.4.3	Sample Reserve Strategy	74
4.5	Performance Evaluation	74
4.5.1	System Identification Problem 1	76
4.5.2	Mackey-Glass Time Series Problem	78
4.6	Summary	80
5	Recurrent Neural Network Meta-cognitive Interval Type-2 Fuzzy Inference System	81
5.1	Introduction	82
5.2	Recurrent Neural Network Meta-cognitive Interval Type-2 fuzzy Inference System	83
5.2.1	Problem Definition	83
5.2.2	Recurrent Neural Network Meta-cognitive Interval Type-2 fuzzy Inference System Architecture	84
5.2.3	Self-regulatory Learning Algorithm for Recurrent Neural Network Meta-cognitive Interval Type-2 fuzzy Inference System	88
5.2.3.1	Sample Delete Strategy	90
5.2.3.2	Sample Learn Strategy	90
5.2.3.3	Sample Reserve Strategy	93
5.3	Performance on Wave Energy Characteristics	93
5.3.1	Waverider Data	93
5.3.2	Performance Measures	97
5.3.3	Significant Wave Height	98
5.3.4	Mean Wave Period	98
5.3.5	Peak Wave Direction	101
5.4	Performance on Wind Speed Prediction Problem	102
5.4.1	RMcIT2FIS for Wind Speed Point Forecast	102
5.4.2	RMcIT2FIS for Wind Speed Prediction Intervals	102
5.5	Summary	104
6	Summary and Future Works	106
6.1	Thesis Summary	107
6.1.1	Meta-cognitive Interval Type-2 Fuzzy Inference System and Its Gradient Descent Learning Algorithm	107
6.1.2	An Evolving Sequential Learning Algorithm for Interval Type-2 Neuro-Fuzzy Inference System	108

Contents

6.1.3	Recurrent Structure for Interval Type-2 Neuro-Fuzzy Inference System	109
6.1.4	Prediction Intervals for Renewable Energy Time-Series Data	109
6.2	Future Directions	110
6.2.1	Improvements in Learning Algorithm	110
6.2.2	Improvements in Renewable Energy Forecasting	111
 Author's Publication		 113
 Bibliography		 114

List of Figures

1.1	Example of point forecasts of wind power generation, Western Denmark.	4
1.2	Example of prediction intervals of wind power generation, Western Denmark.	4
2.1	LUBE structure to generate lower and upper bound.	17
2.2	Gaussian type-1 fuzzy membership function.	19
2.3	Footprint of uncertainty of an interval type-2 fuzzy set.	21
2.4	Categorization of neuro-fuzzy inference systems	23
2.5	Taxonomic timeline of popular neuro-fuzzy systems.	30
3.1	Architecture of the interval type-2 neuro fuzzy system.	37
3.2	Footprint of uncertainty using uncertain means and fixed standard deviation.	39
3.3	McIT2FIS-GD learning mechanism	43
3.4	Training logarithmic prediction error for non-linear system identification problem 2 using McIT2FIS-GD	55
3.5	Actual vs predicted output for non-linear system identification 2 using McIT2FIS-GD	56
3.6	Actual and predicted wind speed for training data using McIT2FIS-GD	58
3.7	Actual and predicted wind speed for testing data using McIT2FIS-GD	59
4.1	Architecture of the interval type-2 neuro fuzzy system.	65
4.2	Footprint of uncertainty using uncertain means and fixed standard deviation.	66
4.3	Actual and predicted output for non-linear system identification problem 1.	77
4.4	Demonstration of Learning Strategy with respect to Instantaneous Error, Adding Threshold and Update Threshold.	78

List of Figures

4.5	Demonstration of Learning Strategy with respect to Spherical Potential and Novelty Threshold.	78
4.6	Actual and predicted output for Mackey-Glass time series problem-85.	79
5.1	Architecture of the recurrent neural network interval type-2 fuzzy inference system.	85
5.2	Footprint of uncertainty using uncertain means and fixed standard deviation.	86
5.3	Directional waverider buoy with integrated high capacity cell and navigation light.	94
5.4	Wave characteristics (Oct 2014 - Nov 2014).	96
5.5	Prediction intervals for significant wave height corresponding to 10% width.	99
5.6	Prediction intervals for mean wave period corresponding to 10% width.	100
5.7	Prediction intervals for wind speed problem.	104

List of Tables

2.1	Summary of major interval type-2 algorithms discussed in literature review.	32
3.1	Performance comparison of McIT2FIS-GD on non-linear system identification problem 1	53
3.2	Performance comparison of McIT2FIS-GD on non-linear system identification problem 2	55
3.3	Performance comparison of McIT2FIS-GD on Mackey-Glass time series problem-85	57
3.4	Performance comparison for wind prediction problem	58
3.5	Experiments on meta-cognitive learning mechanism for wind speed prediction problem	60
4.1	Performance comparison of McIT2FIS-EKF on non-linear system identification problem	76
4.2	Performance comparison of McIT2FIS-EKF on Mackey-Glass time series problem-85	80
5.1	Performance of PIs for significant wave height	99
5.2	Performance of PIs for mean wave period	101
5.3	Performance of PIs for peak wave direction	102
5.4	Performance comparison on wind speed prediction problem	103
5.5	PI evaluation for training and testing	104

Abbreviations

PI	Prediction Interval
ANN	Artificial Neural Network
NFIS	Neuro-Fuzzy Inference System
LUBE	Lower Upper Bound Estimation
TSK	Takagi-Sugeno-Kang
K-M	Karnik-Mendel
ANFIS	Adaptive Network-based Fuzzy Inference System
eTS	Evolving Takagi-Sugeno Model
eT2FIS	Evolving Type-2 Fuzzy Inference System
simpl_eTS	Simplified Evolving Takagi Sugeno Model
SAFIS	Sequential Adaptive Fuzzy Inference System
OS-Fuzzy-ELM	Online Sequential Fuzzy Extreme Learning Machine
SVR	Support Vector Regression
IT2FIS	Interval Type-2 Fuzzy Inference Systems
SIT2FNN	Simplified Interval Type-2 Fuzzy Neural Network
SONFIN	Self-Constructing Neural Fuzzy Inference Network
SEIT2FNN	Self-Evolving Interval Type-2 Fuzzy Neural Network
McFIS	Meta-cognitive Fuzzy Inference System
McIT2FIS	Meta-cognitive Interval Type-2 Fuzzy Inference System
McIT2FIS-GD	McIT2FIS-Gradient Descent

Abbreviations

McIT2FIS-EKF	McIT2FIS-Extended Kalman Filter
RMcIT2FIS	Recurrent Neural Network McIT2FIS
RMSE	Root Mean Squared Error
PS	Percentage of Samples Employed
SD	Standard Deviation
NDEI	Non Destructive Error Index
PICP	Prediction Interval Coverage Probability
PIAW	Prediction Interval Average Width

Symbols

n	number of input features
M	number of output features
$\mathbf{x}$	input feature vector
$\mathbf{y}$	output feature vector
m_1	left center of a fuzzy rule
m_2	right center of a fuzzy rule
σ	width of a fuzzy rule
K	total number of rules
μ^{up}	upper membership strength
μ^{lo}	lower membership strength
w_l, w_r	weight parameters
q_l, q_r	control factors
ψ	spherical potential
$\boldsymbol{\theta}$	parameter vector
E	prediction error
E_d	sample delete threshold
E_a	self-adaptive sample addition threshold
E_u	self-adaptive sample update threshold
E_S	novelty threshold
PI_d	prediction interval delete threshold

Abbreviations

PI_a	prediction interval addition threshold
PI_u	prediction interval update threshold
η	learning rate
κ	overlap factor
γ	slope parameter
p_0	initial error covariance
q_0	initial process noise
r_0	initial measurement noise
$\mathbf{G}$	Kalman gain matrix
$\mathbf{P}$	error covariance matrix
$\mathbf{H}$	gradient matrix with respect to parameters
α_l, α_r	adaptive memory parameters
β_l, β_r	weight parameters of memory signals

Chapter 1

Introduction

This chapter first describes the importance of renewable energy such as wind, solar, wave, tides and their crucial impact on the environment compared to fossil fuel. This chapter then shows the need for accurate forecasting in order to integrate these sources into the power grid and manage the power system effectively. Point forecast and prediction interval are also discussed in this chapter. Next, major issues and challenges of forecasting method are reviewed, followed by the research objectives, research contributions and thesis organization.

1.1 Background and Motivation

Renewable energy is becoming more acceptable as an alternative source of energy. Several commonly used renewable energy sources are biofuels, wind and solar energy in addition to less widely-employed sources such as wave and tides, etc. Contrary to fossil fuels, renewable energy sources regenerate and they provide environmental friendly and natural power, which helps drive down greenhouse gas emissions. Although coal, oil, gas and other fossil fuels still make up approximately three-quarters of final energy consumption, the power generation by renewable energy sources in the next coming decades has been projected to follow a rising trend ¹. Hence, there is a need to develop effective ways to exploit the sources for a sustainable future.

Regarding the integration of green energy into the power grid, the electrical power generation is becoming more complex [1]. One of the main reasons for the complexity is the inability to accurately predict the strength of these sources at a given time. The renewable sources depend heavily on nature, for instance, wind and solar energy are only available when the wind blows and the sun shines; clouds and the low wind shall reduce electricity from solar power plants and wind farms. As inaccurate generation caused by the uncertainty could lead to financial losses, the realistic forecast of the sources is needed for increased and improved renewable energy usage. The process of forecasting can be categorized as long-term, medium-term and short-term. While long-term forecasts and medium-term forecasts play a key role in capacity planning and maintenance scheduling [2, 3], short-term forecasts are the basic component in the control and day-to-day operations of the system. Short-term forecasts enable ones to control the operation of the power system, reduce fuel consumption and increase the reliability [4, 5]. The traditional method of forecasting is based on point forecast, in which only one crisp target value is provided. Since the source is uncertain due to various factors, it is desirable to predict the lower as well as the upper limit of the future

¹<https://www.statista.com>

target. This helps the network operators and decision-makers in scheduling and operating the system more optimally. Therefore, employing an appropriate forecast in a well-defined problem can significantly improve the performance of the power system while considering the risk as well as controlling the losses.

Forecasting is understood as making a statement of the future events based on what has been observed and essential to the integration of renewable energy in the power system. For renewable energy forecasting, even though locations and energy forms might be considered to predict, the temporal dimension is the foremost important one [6]. There are a number of forecasting approaches, e.g., forecasting-based on expert judgment or model-based approach, etc. While expert judgment is based on intuitive observations and empirical experiences of the forecasters, model-based approach is developed with statistics and comprises mathematical representations. For renewable energy, model-based forecasting outperforms expert experience due to the intermittent and uncertain nature of the source. A model-based forecast in the temporal dimension assuming ourselves at time t and considering the future point $(t + \Delta t)$ is given as:

$$\begin{aligned}\hat{y}(t + \Delta t) &= f(X, \theta) \\ \Delta t &= 1, 2, \dots, T\end{aligned}\tag{1.1}$$

,where X is the total observed information up to time t , f is the model and θ is set of parameters. In the equation (1.1), Δt is referred to as the forecasting horizon, the symbol $\hat{y}$ indicates the estimation of the future target. The forecast length T denotes the temporal resolution of the forecasts and in practical T varies in the range from extremely short-term (T is in seconds) to long-term (from one day to one week), depending on the purposes of the forecasts. Recent studies have shown their feasibility in long-term forecasts for wind and solar energy [7–9]. The short-term forecasts have also been studied in [4, 10]. However, the research works on

FIGURE 1.1: Example of point forecasts of wind power generation, Western Denmark.

FIGURE 1.2: Example of prediction intervals of wind power generation, Western Denmark.

other emerging renewable energy sources such as wave, biofuels or geothermal energy are still an underexplored area in the existing literature[6, 11, 12].

There are two major types of renewable energy forecasts, namely, point forecasts and probabilistic forecast. While point forecast emphasizes in the single-value observation of the future event, probabilistic forecast such as prediction interval gives information about potential future outcomes. Prediction intervals provide the level of uncertainty for the coming period by defining a range of future observations with a certain probability. An example of point forecast of wind power generation compared to prediction intervals is shown in Figure 1.1 and 1.2². Figure 1.1 is the point forecasts of wind power generation issued on the 4th April 2007 at 00:00 UTC for the whole onshore capacity of Western Denmark. Point forecast informs the decision-makers that the expected wind power generation at a specific time in the coming hours is a certain value. Figure 1.2 shows prediction intervals of wind power generation with a nominal coverage rate of 90% issued in Denmark at the same time. Prediction intervals give a range of values for every lead time and have an advantage of providing better information, which is extremely useful for the forecasters in scheduling and managing the energy system. Current state-of-the-art model-based forecasting approaches that address prediction intervals have been developed in [10, 13, 14]. Here, multiple networks are used to predict the interval. Further, these intervals are not known and hence, developing supervised learning algorithms is a challenging topic. Recently, it has been shown that the interval type-2 fuzzy sets are capable of modeling the uncertainty and provide the upper and lower bound on the data [15–21]. This motivated the research to build an interval type-2 fuzzy inference system for predicting the forecast intervals, which will provide an approach to quantitatively measure the forecasting outcomes for an efficient energy system.

This thesis deals with prediction intervals and proposes an effective model-based approach for renewable energy forecasting. In the next section, the issues and

²Figure 1.1 and 1.2 are adopted from [6]

challenges in the research are outlined.

1.2 Issues and Challenges

In this section, some major issues and challenges arising in effective prediction of renewable sources are demonstrated in addition to issues associated with the current generation of interval type-2 fuzzy inference systems.

- **Handling Non-stationarity:**

The renewable energy sources are fundamentally non-stationary due to their variable statistical properties and uncertainty in nature. Handling non-stationarity is one of the foremost important tasks in order to effectively integrate the sources into the power system. The uncertainty in wind data, for instance, arises because wind is intermittent and very volatile. Wind power depends on many external uncontrollable factors such as wind speed, air density, turbine characteristics. All of these factors also change depending on the location of the site. Likewise, solar, wave energy and other renewable sources are non-stationary, which calls for a need to develop a system that models the non-stationarity in the data and efficiently predicts the source.

- **Quantifying Uncertainty:**

Handling or modeling uncertainty is itself insufficient for its effective decisions in scheduling and management of the system, thus there is a need to quantify this uncertainty such as providing the mean and variance. Such a quantification of uncertainty might help the decision-makers in efficiently operating the system. The quantitative uncertainty could be measured by determining how likely the targets are given the uncovered system parameters. Even though a set of algorithms in the literature is able to quantify the uncertainty in data to a certain extent, the performances are affected

significantly when it comes to intermittent sources such as wind and wave. Hence, there is a need to develop a method to overcome this problem.

■ **Sequential Learning Framework:**

The data being studied in various practical problems are streaming data with the continually varying values. The training data arrive in sequence and the network has to adjust and adapt the parameters according to the variation of the data. Hence, there is a need to develop an adaptive sequential learning algorithm which learns the sample and evolves its structure based on the data. Moreover, since the data arise one by one, the proposed algorithm should be able to select appropriate learning strategies that over-training are avoided.

1.3 Research Objectives

In this thesis, a data-driven, self-regulate interval type-2 neuro-fuzzy inference system handling prediction intervals is proposed for efficient and realistic prediction of wind and wave characteristics. The research addresses the above-mentioned issues and challenges and aims to develop a system that is able to forecast the renewable sources effectively. The work further discusses the applicability of the system in forecasting. The objectives of this research are summarized below:

- **Development of an Algorithm to Handle Non-stationarity in Data:** To develop an interval type-2 neuro-fuzzy inference system which could model uncertainty in the time series data. The algorithm should be able to handle non-stationarity by employing interval type-2 fuzzy sets. In addition, computational effort in type-reduction of the algorithm should be reduced and the algorithm could be able to evolve for adaptive learning. The learning algorithm would be employed to solve function approximation problems and real-world wind prediction problem.

- **Self-regulation learning mechanism:** To develop a learning mechanism that incorporates self-regulation and adapts the network based on the novelty in data. The sample should be monitored and learning strategy should be controlled in a self-regulatory manner.
 - **Evolving structure:** To develop an algorithm to cope with temporally varying data. The algorithm should enable the network structure to automatically evolve. Such self-evolving capability would solve the problem associated with adaptive learning and initialize the self-learning mechanism in the machine learning framework.
 - **Computationally Efficient Type-Reduction:** To develop a fast data-driven mechanism to simplify the learning process and reduce computational cost. The interval type-reduction process in the system should capitalize straightforward control factors instead of iterative burden method to overcome the time-consuming issue.
- **Development of a Sequential Learning Algorithm for Interval Type-2 Fuzzy Inference System:** To develop a purely sequential learning algorithm for interval type-2 neuro-fuzzy inference system, where data is presented once and discarded after learning. The algorithm should be able to evolve and adapt its parameters based on the data. Such a system will help in handling streaming data. To quantify the uncertainty, the system could provide prediction intervals, which would be an advantage for efficient forecasting.
- **Fully sequential learning algorithm:** To develop a fully online learning algorithm. The algorithm should be able to handle samples arriving sequentially and be critical in the training process to select the appropriate sample for learning. Moreover, the learning algorithm would employ a fast parameter learning scheme to improve the performance of the system.
 - **Adaptive Framework:** To develop an algorithm that is able to adapt the network and handle the changes in the learning environment.

- **Development of a Recurrent Neural Network for Interval Type-2 Fuzzy Inference System:** To develop a learning algorithm that enables associative memories to better approximate the system. The developed algorithm might be able to self-regulate its knowledge in a way such that the parameters of the algorithm are adaptive based on the novelty present in the sample. Further, the learning algorithm could better investigate the uncertainty by employing a measure for prediction intervals. The system should also be capable of quantifying uncertainty and provide a competitive performance.
 - **Uncertainty Quantification:** To develop a system that measures and quantifies the uncertainty associated with the data. The approach should be able to provide a good quantitative measurement as prediction intervals for the underlying non-stationarity of the data.
 - **Wave and Wind Prediction:** To conduct experiments on the wave and wind energy data. The experiments should provide an accurate prediction of wave and wind energy characteristics.

1.4 Research Contributions

The major contributions in the thesis are as follows:

- **Meta-cognitive Interval Type-2 Fuzzy Inference System-Gradient Descent:** In this thesis, an interval type-2 neuro-fuzzy inference system with a fast interval type-reduction and its meta-cognitive learning algorithm is proposed to handle the non-stationarity in data. The system models input features as uncertainty by employing interval type-2 sets in rule antecedent. The rules are of Takagi-Sugeno-Kang type and the learning algorithm of the system uses first-order gradient descent approach. The learning mechanism of the proposed system incorporates self-regulation on an inference mechanism which employs a computationally efficient interval type-reduction. Experimental results indicate that the proposed algorithm performs better

than other well-known type-1 and type-2 fuzzy inference systems in the literature.

- **Self-regulation in Learning:** The learning mechanism employs self-regulatory meta-cognitive learning to evolve the structure and parameters of the network based on the novelty of the training sample. Samples with novel knowledge are learnt first and the ones with low information content should be used to fine-tune the network.
- **Fast Interval Type-Reduction:** A computationally efficient type-reduction algorithm is introduced. The use of this type-reduction technique facilitates the inference mechanism and improves its performance.

■ **Sequential Learning Algorithm for Interval Type-2 Fuzzy Inference**

System: A fully sequential learning algorithm to handle and quantify uncertainty is proposed. A five-layered modified Takagi-Sugeno-Kang interval type-2 fuzzy inference mechanism forms the structure, and the learning algorithm is a self-regulate learning mechanism. It employs an extended Kalman filtering-based method to adapt the parameters of the network. The experiment studies show that the proposed algorithm outperforms other popular type-1 and type-2 fuzzy systems.

- **Sequential learning algorithm:** The samples arrive sequentially and the learning algorithm decides the appropriate strategy for learning, which are rule growing/parameter update or deleted without learning or reserved for a later stage.
- **Regularization of extended Kalman filtering Method:** The extended Kalman filtering is employed to compute the parameter of the network by minimizing the sum of square error. It is computationally efficient and enables online learning mechanism of the network.

- **Recurrent Neural Network Meta-cognitive Interval Type-2 Fuzzy Inference System:** A recurrent neural network meta-cognitive interval type-2 fuzzy inference system is proposed to improve the forecasting outcome. The architecture of the proposed system incorporates two feedback neurons in the output layers making associative memories function. A novel prediction intervals criterion of rule learning strategies is proposed for the system to accurately decide rule growing/parameter update or rule discarding or being reserved to be learnt at the later stage. The system is able to quantify uncertainty as well as provide a good prediction. The experiments on prediction intervals are conducted with wave and wind energy characteristics. The performance evaluation delivers a concrete answer for prediction intervals in online real-time situations through the algorithmic development of a meta-cognitive interval type-2 neuro-fuzzy inference system.

1.5 Thesis Organization

The organization of this report is shown as followings:

- **Chapter 2:** The literature of existing forecasting approaches are reviewed followed by the lower upper bound estimation method for prediction intervals. Existing neuro-fuzzy inferences systems and their learning algorithm are also mentioned and categorized.
- **Chapter 3:** This chapter discusses the learning algorithm to handle the non-stationarity in time-varying data. A self-regulate learning algorithm is also applied together with fuzzy neural network, enabling the system to evolve its parameters and structure based on the data. The system employs a data-driven type-reduction and learns the samples using gradient descent algorithm. Self-regulate learning mechanism controls *what-to-learn*, *when-to-learn* and *how-to-learn* efficiently. The proposed system is evaluated using a real-world wind speed prediction problem and a set of three popular

benchmark system approximation problems. The experimental results are compared and discussed using root mean square error of the training and testing data. The comparable results indicate that the system successfully approximates the functional relationship between the input and the output while employing a fewer number of rules than other neuro-fuzzy inference systems.

- **Chapter 4:** In this chapter, a meta-cognitive sequential learning algorithm for interval type-2 fuzzy inference system for prediction intervals is proposed. The structure of proposed algorithm is a five-layered Takagi-Sugeno-Kang interval type-2 fuzzy inference mechanism. The system is capable of handling sequentially arriving data, and a self-regulate learning mechanism with fast parameters updating algorithm is employed to enable the system to evolve and learn effectively. The performance of the proposed system is compared with a set of neuro-fuzzy systems on benchmark function approximation problems. The results highlight advantages of sequential learning algorithm over other state-of-the-art algorithms in the literature.
- **Chapter 5:** This chapter presents a memory structure for interval type-2 fuzzy neural network. The functional use of the proposed memory structure is associative memories, which is able to help improve the learning quality by memorizing. The learning algorithm of the system is enhanced with a prediction interval measure in deciding appropriate rule learning strategy. The performance of the system is evaluated on wave and wind energy characteristics prediction intervals problem. The wave measurement data was collected from the offshore Semakau Island, Singapore using directional GPS Waveriders over the period of one year. Four wave characteristics were extracted for the study in this chapter. Prediction intervals for significant wave height, mean wave period, peak wave direction and wind speed are constructed and evaluated based on training/testing accuracy, the overall width and the coverage of the intervals. It could be observed that the

algorithm provides a good quantitative measurement of uncertainty for the wave and wind energy data.

- **Chapter 6:** This chapter summarizes the conclusion of the research works which have been studied in this thesis, followed by an outline of potential future directions.

Chapter 2

Literature Review

In the previous chapter, it has been mentioned that forecasting is an important research topic in renewable energy due to its impacts on scheduling and managing the power system. Issues and challenges associated with the topic have also been understood. In this chapter, popular forecasting models in the literature are presented. Further, artificial intelligence models and their existing algorithms for prediction intervals are discussed, followed by various hybrid neuro-fuzzy inference systems and their existing drawbacks.

2.1 Introduction

Renewable energy forecasting is a critical issue in the management of the electrical power systems and can be formulated as a prediction problem in machine learning framework. Conventional forecasting methods for renewable sources such as wind energy can be classified into three approaches: physical model, statistical model and artificial intelligence model [22]. Forecasting using physical models requires physical sensors, satellites or meteorological instruments to predict the source at a particular location. The measured data are then modeled based on the physical dynamics of the sources. The results are accurate but the cost of computation as well as hardware devices is an issue. In case of statistical approaches, some popular models are auto-regressive integrated moving average (ARIMA), and exponential smoothing (ES) models [23, 24]. Although these models have an advantage of being computationally inexpensive, they are only suitable for very-short-term forecast due to their linearity. Within the scope of this thesis, forecasting methods using artificial intelligence models will be discussed. In artificial intelligence community, researchers aim to develop the machines with the human ability to learn and reason. Some typical computing models include artificial neural networks (ANN) [25, 26], fuzzy logic system [27, 28] and the class of their combination, namely hybrid neuro-fuzzy inference systems (NFIS). In the literature of renewable energy forecasting employing artificial intelligence models, research works have been carried out to quantify the associated uncertainty in the generated forecasts [10, 14, 29, 30]. One of the most significant developments in this direction is predicting the lower/upper bound (LUBE) [31]. The method addresses probabilistic forecasts by construction of prediction intervals, wherein the intervals are denoted by the lower and upper estimation. Prediction intervals can appropriately handle the uncertainty associated with renewable energy forecasts and have been studied with the wind data in [14, 29]. In the following section, the LUBE method and its variations are discussed followed by the literature of different neuro-fuzzy inference systems available, along with their limitations.

2.2 Lower Upper Bound Estimation Method

As mentioned earlier, point forecast is currently a fundamental way of forecasting and has most of the applications. However, scheduling and management concern not only the exact value but the upper and lower in the target. As a result, prediction interval is preferred over point forecast. In the literature, different methods for constructing prediction intervals have been used. Delta technique interprets a neural network as a non-linear regression model and linearizes them based on Taylors series expansion [32, 33]. Bayesian technique obtains probability distributions of the target by integrating them on the observed training set [34]. Bootstrap is a resampling method [35] and mean-variance estimation develops two neural networks for prediction of mean and variance of targets [36]. Because of the massive implementation and data distribution assumption, the mentioned traditional methods are restricted in applications. The lower upper bound estimation (LUBE) method for prediction intervals has been proposed to overcome these above problems [31]. The performance of LUBE method has been revealed in various prediction problems such as load forecasting [37], travel time prediction [38], short-term electrical power forecasting [39] and wind power generation forecasting [14]. In [13, 14], the lower upper bound estimation method is extended with particle swarm optimization (PSO) in the learning phase to construct prediction intervals. The structure of the method is demonstrated in Figure 2.1. The structure utilized in LUBE method is a fully connected feed-forward neural network, and the network comprised of four layers: input layer, two hidden layers and output layer. The activation functions in the hidden and output layers are *tansig* and *purelin* respectively. The functions of each layer resemble the ones in a typical feed-forward artificial neural network [25]. The number of neurons in each layer is determined by a validation data set. The weights connecting the hidden layer with input layer and the output layer are represented as parameters of PSO. In order to construct optimal prediction intervals, PSO searches for the best candidate in the population based on an objective function. The objective function is formulated from two main criteria: the width of the interval and the

FIGURE 2.1: LUBE structure to generate lower and upper bound.

coverage probability that predicted values will fall into the intervals [13, 14, 39]. Prediction intervals then can be constructed in an easy and straightforward way.

Although LUBE method is simpler, faster and more reliable than the traditional methods, its limitations include fixed architecture and time-consuming PSO-based searching capability. PSO-LUBE is unable to handle the non-stationarity which is inherent in wind or wave data. Hence, there is a need to develop a system which can handle the non-stationarity as uncertainty in data and adapt its structure based on the data. In the next section, a detailed review of another class of artificial intelligence models applicable for renewable energy forecasting called hybrid neuro-fuzzy system is presented.

2.3 A Review of Hybrid Neuro-Fuzzy Inference System.

Artificial Neural Network (ANN) is the biologically inspired simulations of the human brain [25]. It resembles the brain in the way that ANN acquires knowledge through learning and ANN stores and increases knowledge by neuron connections denoted as synaptic weights. On-going researches have been conducted on variations of ANN and applied to renewable energy forecasting in the literature [7, 40–46]. In [40], the wind power forecast was produced by a modified ANN. Recurrent ANN was used for time-series forecasting in [7] and in [42] an optimization method was applied to tune the parameters of a hybrid neural network for wind power prediction. Solar power forecasts were produced by an ANN model in [43]. Studies on wave prediction using artificial neural networks have also been conducted in [44–46]. Research works have been carried out to combine artificial neural networks with fuzzy logic to exploit the capability of the mimic human-reasoning fuzzy inference. The combination is called hybrid neuro-fuzzy inference system and has been discussed in [47, 48].

Fuzzy logic theory as one of the most influential developments which mimic human-like thinking was first introduced by Zadeh in [27]. It uses fuzzy rules and fuzzy inferences to enable the system to tolerate imprecision and uncertainty. The membership function of a fuzzy set makes linguistic variables easy to be described. Fuzzy theory is different in the way that fuzzy logic takes values of a variable from zero to one, hence enables computer to mimic human linguistics. Because the initial type-1 fuzzy logic gives a crisp membership value for every input, it is unable to handle uncertainty. Zadeh has proposed type-2 fuzzy sets in [28] and type-2 fuzzy logic systems in [49, 50]. Type-2 fuzzy sets allow researchers to model uncertainty, but they are computationally expensive. Therefore, interval type-2 fuzzy sets have been introduced in [51]. The details of type-1 and type-2 fuzzy sets are described in the next subsections.

FIGURE 2.2: Gaussian type-1 fuzzy membership function.

2.3.1 Type-1 Fuzzy Set

A type-1 fuzzy set, A , using a single variable, $x \in X$, can be represented as:

$$A = \{(x, \mu_A(x)) \mid \forall x \in X\} \quad (2.1)$$

A can also be defined as:

$$A = \int_{x \in X} \mu_A(x) / x \quad (2.2)$$

The membership function of a Type-1 fuzzy set is $\mu_A(x)$. This function is a two-dimensional function with values between 0 and 1 for all $x \in X$. The Gaussian Type-1 fuzzy membership function is shown in Figure 2.2. Type-1 fuzzy is unable to handle uncertainty because one crisp membership is associated with each input data.

2.3.2 Type-2 Fuzzy Set

A Type-2 fuzzy set $\tilde{A}$ with a type-2 membership function $\mu_{\tilde{A}}(x, u)$ is defined as:

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (2.3)$$

where, $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. $\tilde{A}$ can also be defined as:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0, 1] \quad (2.4)$$

Type-2 fuzzy sets are utilized by researchers to model uncertainty, but due to their computationally expensive, interval type-2 fuzzy sets are proposed to overcome the issue [51].

2.3.3 Interval Type-2 Fuzzy Set

$\tilde{A}$ as an interval type-2 fuzzy set is defined as:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} 1 / (x, u) \quad J_x \subseteq [0, 1] \quad (2.5)$$

The difference compared to a type-2 fuzzy set is that all $\mu_{\tilde{A}}(x, u)$ equal to 1 for an interval type-2 fuzzy set. This assignment makes the computation easier hence the fuzzy system becomes simpler and still handles uncertainty. An example of the footprint of uncertainty (FOU) of interval type-2 fuzzy set is shown in Figure 2.3. The proposal of interval type-2 fuzzy system has been broadly implemented and integrated in the realm of neuro-fuzzy system, and it has been shown to be an effective avenue to improve the predictive performance of a neuro-fuzzy system.

FIGURE 2.3: Footprint of uncertainty of an interval type-2 fuzzy set.

This class of algorithms is known as Interval Type-2 Neuro-Fuzzy Inference Systems (IT2NFIS or IT2FIS) and has been widely developed in the literature [15–21, 52–58]. Hybrid neuro-fuzzy inference systems can be mainly classified based on several related criteria including types of fuzzy inference mechanism, types of membership function, types of learning mode and types of application. The categorization of NFIS is shown in detail in Figure 2.4. Some of the categories are described below:

- Classification based on types of fuzzy inference mechanism:

- Takagi-Sugeno-Kang fuzzy inference mechanism [59].
- Mamdani fuzzy inference mechanism [60].
- Others.

- Classification based on types of learning mode:

- Batch/ Offline learning.
- Sequential/ Online learning.

- Classification based on types of modelled uncertainty:
 - Membership function with uncertain mean.
 - Membership function with uncertain standard deviation.
- Classification based on types of application:
 - Classification Type.
 - Function Approximation Type.

The next section discusses on different existing hybrid neuro-fuzzy inference systems in major categories, along with their features as well as limitations.

2.3.4 Classification Based on Types of Fuzzy Inference Mechanism

The two main fuzzy inference mechanisms broadly used in the literature are Takagi-Sugeno-Kang inference mechanism [59] and Mamdani fuzzy inference mechanism [60]. They are discussed in the following section.

2.3.4.1 Mamdani Fuzzy Inference Mechanism

Mamdani fuzzy inference mechanism was proposed by E. H. Mamdani in [60]. The main advantage of this inference mechanism is its rule interpretability. According to the study, the k^{th} rule R_k of the system is expressed as:

$$\begin{aligned}
 R_k : & \textbf{IF } x_1 \text{ is } \tilde{A}_{1k} \text{ AND } x_2 \text{ is } \tilde{A}_{2k} \text{ AND } \dots \text{AND } x_n \text{ is } \tilde{A}_{nk} \\
 & \textbf{THEN } y_j = \tilde{B}_{jk}
 \end{aligned} \tag{2.6}$$

where, $\mathbf{x} = [x_1, x_2, \dots, x_n] \in R^{1 \times n}$ is n -dimensional input and $\mathbf{y} = [y_1, y_2, \dots, y_M] \in R^{1 \times M}$ is M -dimensional output. $\tilde{\mathbf{A}}_k = [\tilde{A}_{1k}, \tilde{A}_{2k}, \dots, \tilde{A}_{nk}] \in R^{1 \times n}$ are the fuzzy rule antecedents of k^{th} rule and $\tilde{B}_{jk}$ is the corresponding consequent.

FIGURE 2.4: Categorization of neuro-fuzzy inference systems

The membership of i^{th} input feature of a sample x with respect to k^{th} rule is given by:

$$\phi_{ik} = \exp\left(-\frac{(x_i - m_{ik})^2}{2(\sigma_k)^2}\right), \quad i = 1, 2, \dots, n \quad (2.7)$$

where, m_k and σ_k are the center and width of k^{th} Gaussian rule. Other membership functions such as triangular, trapezoidal membership functions can also be employed instead of using the Gaussian membership function for fuzzification. The aggregation of all the computed memberships is given as:

$$\begin{aligned} \phi_K &= T - \text{norm}(\phi_{1k}, \dots, \phi_{nk}) \\ &= \min(\phi_{1k}, \dots, \phi_{nk}) \end{aligned} \quad (2.8)$$

Here, $T - \text{norm}$ operator is employed for feature aggregation and a $S - \text{norm}$ operator is used to compute the membership of the aggregated rule antecedents to the output. Research works employing Mamdani fuzzy inference mechanism in the literature of type-1 and type-2 neuro-fuzzy inference systems are available in [19, 61–63].

2.3.4.2 Takagi-Sugeno-Kang Fuzzy Inference Mechanism

Takagi-Sugeno-Kang Fuzzy Inference Mechanism (TSK) in [59] expresses the k^{th} rule R_k as:

$$\begin{aligned} R_k : & \text{IF } x_1 \text{ is } \tilde{A}_{1k} \text{ AND } x_2 \text{ is } \tilde{A}_{2k} \text{ AND } \dots \text{AND } x_n \text{ is } \tilde{A}_{nk} \\ & \text{THEN } y_j = \tilde{a}_{0jk} + \tilde{a}_{1jk}x_1 + \dots + \tilde{a}_{njk}x_n \end{aligned} \quad (2.9)$$

where, $\mathbf{x} = [x_1, x_2, \dots, x_n] \in R^{1 \times n}$ is n -dimensional input and $\mathbf{y} = [y_1, y_2, \dots, y_M] \in R^{1 \times M}$ is M -dimensional output. $\tilde{\mathbf{A}}_k = [\tilde{A}_{1k}, \tilde{A}_{2k}, \dots, \tilde{A}_{nk}] \in R^{1 \times n}$ are the fuzzy rule antecedents of k^{th} rule and $\tilde{a}_k$ are the consequent parameters of the k^{th} rule. In the fuzzy inference systems employing Takagi-Sugeno-Kang mechanism, the output is a weighted combination of inputs and this mechanism can model a system with accuracy and fewer number of rules. The membership of i^{th} input feature of a sample $\mathbf{x}$ with respect to k^{th} rule is given by:

$$\phi_{ik} = \exp\left(-\frac{(x_i - m_{ik})^2}{2(\sigma_k)^2}\right), \quad i = 1, 2, \dots, n \quad (2.10)$$

where, m_k and σ_k are the center and width of k^{th} Gaussian rule. In the next step, aggregation of features using a product of membership functions is given as:

$$\begin{aligned} \phi_K &= T - norm(\phi_{1k}, \dots, \phi_{nk}) \\ &= \phi_{1k} \times \dots \times \phi_{nk} \end{aligned} \quad (2.11)$$

The output of a Takagi-Sugeno-Kang fuzzy inference system is computed by a weighted center of gravity defuzzified outputs and is given by:

$$\hat{y} = \frac{\sum_{k=1}^K v_k \cdot \phi_k}{\sum_{k=1}^K \phi_k} \quad (2.12)$$

Studies on type-1 and type-2 neuro-fuzzy inference systems employing TSK mechanism are available in [18, 20, 64–67]. In the literature, other inference mechanisms have also been proposed [68–70]. A heuristic fuzzy logic rule and input-output membership function for time-series and regression problems have been proposed in [68]. The rule antecedent and consequent employ $T - norm$ and $S - norm$ operator to improve the accuracy and interpretability of the system. In [69, 70], the combination of Mamdani and TSK mechanism in the rule consequent

has been studied to attain improvement of a fuzzy inference system. In such a system, two rule consequents are computed regarding Mamdani and TSK inference mechanism. The overall output is computed by combining both inferred outputs to achieve better accuracy. In the next section, different neuro-fuzzy inference systems are categorized by their types of learning mode.

2.3.5 Classification Based on Types of Learning Mode

Fuzzy systems and artificial neural networks have the ability to estimate any non-linear function, provided that a sufficient number of rules or neurons are available. Neuro-fuzzy inference systems (NFIS) understood as the unification of these technologies. They have been shown to have good interpretability and prediction ability [65, 66, 71–81]. In this section, NFIS are classified based on the types of employed learning mode which include batch/offline learning and sequential/online learning.

2.3.5.1 Batch/Offline Learning

In offline learning algorithms, the training data is assumed to be available *a priori* while in batch learning, data arrives in batch and the network adapts its parameters based on the data. In these types of learning mode, the network structures are normally fixed and the rule initialization is based on clustering algorithms. In the literature, various type-1 and type-2 fuzzy inference systems employ a batch/offline learning mode [16, 17, 53–55, 71, 82, 82–84].

Some of the initial works in type-1 neuro-fuzzy inference systems employing such a learning mode are neural-network-based fuzzy logic control and decision system (FALCON) [82], adaptive neuro-fuzzy inference system (ANFIS) [71], evolving Takagi-Sugeno fuzzy model [83]. In [71], a Takagi-Sugeno-Kang (TSK) based adaptive fuzzy inference system with a five-layer network structure has been developed. It employs an iterative back propagation learning algorithm to

update the antecedent parameters and least mean square estimation to determine the antecedent parameters. In [82], FALCON is formulated by a feed-forward neural network with five layers. The combination of unsupervised and supervised learning in FALCON has improved convergence speed and helped avoid rule-matching time. In [83], a neuro-fuzzy architecture has been extended for both function approximation and classification purposes. The model is called NEFPROX and employs a supervised learning mechanism. Various neuro-fuzzy inference systems which use evolutionary algorithms have been proposed in the literature [75, 85–87]. Evolutionary algorithm in [75, 85] is utilized for structure identification while the algorithm helps in choosing membership functions, learning parameters and the optimal learning rate of the network in [86]. Since these algorithms generate large population for finding optimal parameters, they are considered as employing an offline scheme.

In various practical problems, the data are non-stationary and the network needs an approach to model and cope with that non-stationarity. The above mentioned neuro-fuzzy inference systems are unable to efficiently handle the uncertainty in data due to the precise feature of type-1 fuzzy sets. As mentioned in the previous section, IT2FIS employing interval type-2 fuzzy sets have the capability of modeling uncertainty and a class of IT2FIS has been proposed in the literature. In the early studies of interval type-2 neuro-fuzzy inference systems, fixed architectures have been applied and the learning algorithms focus on that architectures. The class of these IT2FISs is based on offline/batch learning mode, in which the learning algorithms use samples arriving in batch to update the parameters of the system. In the literature, various studies employ offline/batch learning for interval type-2 fuzzy systems [16, 17, 53–55, 84]. The structures of these systems are normally static, and the rule initialization are based on a clustering algorithm. In [55], three interval type-2 fuzzy neural networks (IT2FNNs) have been proposed using Takagi-Sugeno-Kang inference mechanism. The work has studied the performance of IT2FNN when antecedents and consequents and reduction type in the neural network are integrated in different ways. Throughout the work, the structures of IT2FNNs are assumed to be fixed. The hybrid learning algorithm between

gradient descent and adaptive back-propagation is used to determine the network parameters. In [53], a five-layer neuro-fuzzy inference system integrated with recurrent neural network has been proposed. The network structure is determined based on specific problem and the corresponding learning algorithm is derived by gradient descent method. In [17], an extended Kalman filter based learning algorithm for interval type-2 neuro-fuzzy systems has been proposed. The structure is fixed and the parameters of the network are trained by Kalman filter in a feedback error learning scheme. In [88], an interval type-2 neuro-fuzzy system is presented and the structure selection is derived based on the fuzzy clustering approach. Parameters update employs gradient descent algorithm. The clustering algorithm determines the number of fuzzy rules from the batch of samples, which makes the system offline.

2.3.5.2 Sequential/Online Learning

In the previous section, a class of offline/batch algorithms which adapt the parameters based on static network structures has been reviewed. Static architectures and offline/batch learning are not appropriate for temporally varying data such as wind and wave data [21, 89, 90] since, in order to handle the temporally time-varying data, the systems should have the ability to evolve both their structures and parameters based on the data. A class of evolving and adaptive algorithms for type-1 and interval type-2 fuzzy inference systems has been developed in the literature [18–20, 20, 56, 58, 66, 72–74, 79, 80, 91–95]. In these systems, the networks grow and adapt based on the training data.

Self-organized learning in [72, 96] is one of the first algorithms that employ such type of learning mode. A self-constructing neuro-fuzzy inference network with its Takagi-Sugeno-Kang inference mechanism is proposed in [72]. The system employs a projection based correlation and a recursive least square technique for structure and parameters learning. In [96], a self-constructing neuro-fuzzy inference system has been proposed in which the rules are grown in an online fashion.

A recursive least square method is employed for parameter tuning. However, as the algorithm requires all the training samples for rule pruning, it is considered partially sequential/online. Some of the algorithms were developed with extreme learning machine [58, 80]. In [80], an online sequential fuzzy extreme learning machine has been proposed for function approximation and classification problems. The number of rules in this algorithm has to be initially decided and the rules are tuned based on a recursive least square mechanism. Recently in [91], the extreme learning algorithm has been carried out to address the uncertainty in the data stream with an online feature selection method. The algorithm has focused on classification problems and been an encouraging counterpart of other fuzzy inference systems. In [73], a purely sequential adaptive neuro-fuzzy inference system (SAFIS) has applied the concept of the influence of a rule and an extended Kalman filter to grow and adapt rules. The rules are grown and pruned from the network based on their influences with respect to each sample. The system is a sequential learning algorithm in which samples are presented only once and discarded after learning.

In the domain of interval type-2 fuzzy inference systems, various algorithms that employ gradient descent for learning have been initially proposed [18, 19, 62, 97]. In [18], an IT2FNN which employs control factors in the type-reduction layer and uses gradient descent to train the parameters has been proposed. The system is able to evolve and grow rules automatically. In [19], an evolving interval type-2 fuzzy inference system, or eT2FIS generates a new rule if knowledge in arriving data sample is novel. Gradient descent-based algorithm is used to train the network. The system is online learning but since it employs Mamdani inference mechanism, proper accuracy is not archived. In [97], a reduced interval type-2 neural fuzzy system employs structure learning and gradient descent-based parameters learning to be implemented on a chip for real-time application. Another class of interval type-2 algorithms utilizes Kalman filters with gradient descent to tune the parameters of network [20, 52, 61, 98, 99]. A self-evolving interval type-2 fuzzy neural network (SEIT2FNN) with its online clustering method has been proposed in [20]. It uses an ordered Kalman filter and gradient descent for

consequent and antecedent parameters learning respectively. However, the algorithm requires all samples to train multiple times and is not purely online. In [98], a mutually recurrent architecture has been developed for an IT2FIS, and a recurrent self-evolving network for dynamic system processing has been proposed in [52]. The novel recurrent architectures together with online structures and parameters learning are based on the TSK inference mechanism with interval weights. However, the convergence is time-consuming due to the iterative nature of the methods. A summary of the presented interval type-2 neuro-fuzzy inference systems have been provided in Table 2.1, and a taxonomic timeline showing the progression of popular architectures is demonstrated in Figure 2.5.

FIGURE 2.5: Taxonomic timeline of popular neuro-fuzzy systems.

2.3.6 Summary

In this chapter, a review of forecasting methods for temporally time-varying data such as wind and wave data has been presented. The advantages of probabilistic forecast as prediction intervals over conventional point forecast have been described. The literature of type-1 and type-2 neuro-fuzzy inference systems and their learning algorithms have also been categorized and reviewed.

Existing interval type-2 neuro-fuzzy inference systems lack a mechanism to quantify the uncertainty associated with the data even though literature has shown their ability in handling uncertainty. In renewable energy forecasting, prediction intervals which provide the upper and lower estimation of the future outcomes are the potential option for uncertainty quantification. However, considering that the weakness of current works on prediction intervals is their static architectures, this thesis aims to develop an evolving sequential learning algorithm for interval type-2 fuzzy inference system so that the system is able to handle the non-stationarity in data and self-regulate its knowledge to provide prediction intervals. In the next chapter, an evolving neuro-fuzzy inference system and its gradient descent learning algorithm is presented.

Algorithm	Inference Mechanism	Type of MF	Learning Mode	Type of Consequent
Wang et al. [16]	TSK	UM(Gaussian)	Offline (Genetic Algorithm)	FA
Lin et al. [18]	TSK	UM(Gaussian)	Online(Online Clustering+Gradient Descent)	FA
Juang et al. [62]	Mamdani	UM	Online (Gradient Descent)	FA
Tung et al. [19]	Mamdani	UM(Gaussian)	Online (Online Clustering+Gradient Descent)	FA
Pratama et al. [100]	Hybrid	US	Online	Classifier
Castro et al. [55]	TSK	US	Offline	FA
Cai et al. [101]	TSK	UM	Offline (Genetic Algorithm)	Classifier
Juang and Tsao [20]	TSK	UM	Online (Rule Ordered Kalman Filter)	FA
Das et al. [102]	TSK	UM(Gaussian)	Online (Projection Based)	Classifier
Das et al. [57]	TSK	UM(Gaussian)	Online (EKF Based)	FA
Juang et al. [64]	TSK	UM(Gaussian)	Online (Two-Phase SVR)	FA
Lin et al. [99]	TSK	UM	Online (Rule Ordered Kalman Filter)	FA
Lin et al. [98]	TSK	UM	Online (Rule Ordered Kalman Filter)	FA

*FA: Function Approximation

TABLE 2.1: Summary of major interval type-2 algorithms discussed in literature review.

Chapter 3

Meta-cognitive Interval Type-2 Fuzzy Inference System and Its Gradient Descent Learning Algorithm

In the previous chapter, a review of existing interval type-2 fuzzy inference systems has been provided. This chapter focuses on an evolving meta-cognitive interval type-2 fuzzy inference system and its self-regulatory learning algorithm to solve function approximation problems. The proposed system employs a computationally fast data-driven mechanism for interval type-reduction and a gradient-descent algorithm to adapt the parameters of the network. The performance of proposed system is evaluated on benchmark forecasting data sets and a real-world wind speed prediction problem then is compared with other well-known neuro-fuzzy inference systems in the literature.

3.1 Introduction

It has been shown in the previous chapter that due to the precise nature of underlying type-1 fuzzy sets, type-1 neuro-fuzzy systems do not have the ability to handle noise and uncertainty. In addition, type-2 fuzzy sets proposed in [49] allow researchers to model uncertainty because the primary and secondary membership function of a type-2 fuzzy set is capable of handling uncertainty in the data. However, the usage of type-2 fuzzy set is computationally expensive because of an additional type-reduction operation [103]. In order to remove this computational burden, type-2 fuzzy set has been extended to interval type-2 fuzzy sets [51, 104]. These interval type-2 fuzzy sets handle the noise and uncertainty in data using a bounded interval. Many algorithms based on these interval type-2 fuzzy sets known as interval type-2 neuro-fuzzy inference systems (IT2FIS) have been proposed in literature [16, 18, 19, 53, 55, 57, 64, 84, 92, 97, 100, 105, 106].

In [16, 53, 55, 84] IT2FIS with fixed architecture have been proposed. To handle time-varying data, IT2FIS have been proposed that have the capability to evolve its parameters and structure based on the data. In [64] the structure and parameters have been evolved in two different phases. In the first phase, the structure was evolved, while, in the second phase the parameters were tuned. In [18, 19, 105, 106], the structure and associated parameters were evolved simultaneously. In most of the aforementioned algorithms, the consequent weights need to be reordered using the Karnik-Mendel (K-M) algorithm [107]. However, the iterative nature of K-M algorithm makes the inference computationally expensive [108]. Various algorithms have been proposed in literature with an aim to reduce the computational effort in type-reduction [18, 97]. In [18], a simplified interval type-2 fuzzy neural network (SIT2FNN) which uses control factors to adjust the consequent parameters was proposed. It employs a gradient decent based learning algorithm to adapt the control and other parameters of the network. However, the learning algorithm lacks the ability to self-regulate its knowledge, which might cause over-training and poor generalization. The literature has shown that in the

domain of type-1 neuro-fuzzy inference systems [67, 109] and type-2 neuro-fuzzy inference systems [56, 57, 92], meta-cognition based self-regulation has the ability to generalize well. Hence, there is a need to propose an algorithm which incorporates the principles of meta-cognition to avoid over-fitting and provide better generalization.

In this chapter, an interval type-2 fuzzy inference system with data-driven type-reduction and its meta-cognitive learning algorithm is proposed. The work done in this chapter has been published as an article in [56]. In order to solve approximation problem, the structure of the system employs interval type-2 sets in rule antecedent. The rules are of Takagi-Sugeno-Kang type. A data driven type reduction technique is employed to adjust the contribution of lower and upper rule firing values. The learning starts with zero rules and as a new sample arrives, the learning algorithm evolves the parameters and structure based on knowledge content of the sample. Here, a new rule is added by exploiting the Gaussian localization property [57]. The parameters of the network are adapted employing a gradient descent based learning algorithm. We refer it to as meta-cognitive interval type-2 neuro-fuzzy inference system-gradient descent (McIT2FIS-GD).

The performance of McIT2FIS-GD is evaluated on three benchmark prediction problems and a real-world wind forecasting problem. First, the performance of McIT2FIS-GD is analyzed on non-linear system identification problem 1 [72] followed by system identification problem 2 [73] and Mackey-Glass chaotic time series prediction problem [110]. Finally, its performance is evaluated on a real-world wind speed prediction problem [90]. The prediction of wind speed is necessary for efficient power generation. The performance comparison with existing algorithms like SAFIS [73], self-constructing neural fuzzy inference network [72], evolving Takagi-Sugeno model (eTS) [66], simplified evolving Takagi-Sugeno system (Simple_eTS) [111], a self-evolving interval type-2 fuzzy neural network [20], SIT2FNN [18], reduced interval type-2 neural fuzzy system using weighted bound-set boundaries [97] and evolving type-2 neural fuzzy inference system (eT2FIS) [19] indicates better performance of the proposed method.

3.2 Meta-cognitive Interval Type-2 Neuro-Fuzzy Inference System Gradient Descent

McIT2FIS-GD consists of two components, a neuro-fuzzy inference system and a meta-cognitive learning mechanism which controls the learning process of the neuro-fuzzy inference system. In this section, we first describe the problem definition of the proposed algorithm. The architecture and inference mechanism of the interval type-2 neuro-fuzzy inference system are then described followed by the self-regulate meta-cognitive learning mechanism of McIT2FIS-GD.

3.2.1 Problem Definition

Let us assume that we are given training data, $[(\mathbf{x}(1), \mathbf{y}(1)), \dots, (\mathbf{x}(t), \mathbf{y}(t)), \dots]$, where $\mathbf{x}$ is the input to the network and $\mathbf{y}$ is the corresponding target. Without loss of generality, we assume the system to consist of n input and M output nodes. It has processed $t - 1$ samples and added K rules. Then the input is $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ and the output is given by $\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_M(t)]$. The objective of McIT2FIS-GD is to estimate the underlying functional relationship $f[\cdot]$ such that the predicted output:

$$\hat{\mathbf{y}}(t) = f[\mathbf{x}(t), \boldsymbol{\theta}] \quad (3.1)$$

is as close as possible to the desired output, $\mathbf{y}(t)$. It must be noted that $\boldsymbol{\theta}$ is the parameter vector of McIT2FIS-GD. The error for t -th sample measures the difference between the predicted output and the desired output (error $\mathbf{e}(t) = [e_1(t), e_2(t), \dots, e_M(t)]^T$) and is given as:

$$e_o(t) = y_o(t) - \hat{y}_o(t) \quad o = 1, 2, \dots, M \quad (3.2)$$

3.2.2 Meta-cognitive Interval Type-2 Neuro-Fuzzy Inference System Gradient Descent Architecture

In this section, the detailed architecture of McIT2FIS-GD is provided. The proposed algorithm employs an evolving interval type-2 neuro-fuzzy inference system. The architecture is a five-layer network realizing Takagi-Sugeno-Kang inference mechanism [59]. The network is demonstrated in Figure 3.1. It consists of an input layer, a membership layer, a firing layer, an output processing layer and an output layer. The neurons in membership layer employ Gaussian membership function with uncertain mean and fixed standard deviation to form the rule antecedent of the system. Subsequently, the firing layer calculates the membership strengths of combining rules in the network. The weight parameters and control factors connecting firing layer and output processing layer form the rule consequent of interval type-2 neuro-fuzzy inference system. The detailed description of inference mechanism on presentation of the t -th sample, $x(t)$ is given below:

FIGURE 3.1: Architecture of the interval type-2 neuro fuzzy system.

Layer 1- Input layer: This layer contains n nodes representing n number of input features. The input is passed directly to layer 2, membership layer. The output of j -th node is given:

$$u_j(t) = x_j(t); j = 1, 2, \dots, n \quad (3.3)$$

Layer 2- Membership layer: This layer employs an interval type-2 Gaussian membership function to fuzzify the inputs. The footprint of uncertainty is bounded by upper and lower membership function demonstrated in Figure 3.2. The membership of j - th input feature with i - th rule is given by:

$$\mu_{ij}^{up}(t) = \begin{cases} \phi(m_1^{ij}, \sigma^i, u_j(t)) & u_j(t) < m_1^{ij} \\ 1 & m_1^{ij} \leq u_j(t) \leq m_2^{ij} \\ \phi(m_2^{ij}, \sigma^i, u_j(t)) & u_j(t) > m_2^{ij} \end{cases} \quad (3.4)$$

$$\mu_{ij}^{lo}(t) = \begin{cases} \phi(m_2^{ij}, \sigma^i, u_j(t)) & u_j(t) \leq \frac{m_1^{ij} + m_2^{ij}}{2} \\ \phi(m_1^{ij}, \sigma^i, u_j(t)) & u_j(t) > \frac{m_1^{ij} + m_2^{ij}}{2} \end{cases} \quad (3.5)$$

where,

$$\phi(m^{ij}, \sigma^i, u_j) = \exp\left(-\frac{(u_j - m^{ij})^2}{2(\sigma^i)^2}\right); i = 1, 2, \dots, K \quad (3.6)$$

where m_1, m_2 are the left and right center of the interval type-2 Gaussian membership function accordingly, and σ^i is the width of the i - th rule. The output of this layer can be represented by the intervals $[\mu_{ij}^{lo}(t), \mu_{ij}^{up}(t)]$.

Layer 3- Firing layer: Each node in this layer represents the upper and lower firing strength of a rule. It is calculated by the algebraic product of membership values of a rule. The output of this layer is given as:

$$[f_i^{lo}(t), f_i^{up}(t)]; i = 1, 2, \dots, K \quad (3.7)$$

FIGURE 3.2: Footprint of uncertainty using uncertain means and fixed standard deviation.

where,

$$f_i^{lo}(t) = \prod_{j=1}^n \mu_{ij}^{lo}(t) \quad (3.8)$$

and

$$f_i^{up}(t) = \prod_{j=1}^n \mu_{ij}^{up}(t) \quad (3.9)$$

Layer 4- Output processing layer: This layer has K nodes, each of which represents a rule in the network. Instead of using Karnick-Mendel iterative procedure [107] to find lower and upper endpoints, the control parameters q_l and q_r [18] are employed to increase the learning speed and minimize the computational complexity. The output of the firing layer is combined with consequent weight parameters, and the contributions of rules are adjusted to process the output. The two bounds of o – th output feature at time t – th are $y_{lo}^o(t)$ and $y_{up}^o(t)$. The

formulas to calculate y_{lo} and y_{up} in this layer are given by:

$$y_{lo}^o(t) = \frac{(1 - q_l^o) \sum_{i=1}^K f_i^{up}(t) w_l^{io} + q_l^o \sum_{i=1}^K f_i^{lo}(t) w_l^{io}}{\sum_{i=1}^K (f_i^{lo}(t) + f_i^{up}(t))}; \quad (3.10)$$

$$o = 1, 2, \dots, M$$

$$y_{up}^o(t) = \frac{(1 - q_r^o) \sum_{i=1}^K f_i^{lo}(t) w_r^{io} + q_r^o \sum_{i=1}^K f_i^{up}(t) w_r^{io}}{\sum_{i=1}^K (f_i^{lo}(t) + f_i^{up}(t))}; \quad (3.11)$$

$$o = 1, 2, \dots, M$$

Layer 5- Output layer: The output of this layer is the sum of y_{lo} and y_{up} from output processing layer. It is given by:

$$\hat{y}_o(t) = y_{lo}^o(t) + y_{up}^o(t); \quad o = 1, 2, \dots, M \quad (3.12)$$

Next, the meta-cognitive learning algorithm to estimate the parameters and control rules growing mechanism of the system is presented.

3.2.3 Meta-cognitive Gradient Descent Algorithm for McIT2FIS

A self-regulatory learning algorithm forms the meta-cognitive learning mechanism of the system. During learning, the meta-cognitive component monitors the knowledge in the current sample with the knowledge presented in the network and decides to delete the sample (*sample-delete-strategy*), or learn it (*sample-learn-strategy*) or reserve the sample (*sample-reserve-strategy*). The mechanism is shown in Figure 3.3. When a new sample is presented to the network, meta-cognitive learning mechanism monitors the knowledge in the network with respect to the knowledge in the sample. The learning process is then controlled by

deciding whether the sample is employed for growing a new rule/updating the parameters or being discarded without learning. The knowledge difference between the current sample and the network is calculated using two measures: prediction error and spherical potential [67]. The measures and learning strategies of the algorithm are now discussed in details.

Similar to existing algorithms in the literature of neuro-fuzzy inference systems, the popular error-based criterion used in this thesis to estimate the difference in knowledge of a sample is prediction error. The prediction error for the sample $t - th$ is defined as:

$$E(t) = \frac{1}{2}e^2(t) \quad (3.13)$$

where, $e(t)$ is defined in equation (3.2). In addition to the error-based measure, McIT2FIS-GD employs spherical potential, which is broadly used in kernel methods [112, 113] as a measure of novelty, to make the decision on the contained knowledge in the current sample. In this thesis, it is defined as the average distance of the current sample from contributing rules in the network. Spherical potential has been expressed based on the projection of an input sample ($\mathbf{x}$) on to a hyper-dimensional feature space. In this system, the center ($\mathbf{m}$) and the width (σ) of the Gaussian rule antecedents describe the hyper-dimensional feature space. Let the center of the K -dimensional space be Φ_0 , the spherical potential of any sample ($\mathbf{x}$) [113] is expressed as:

$$\psi = \|\Phi(\mathbf{x}) - \Phi_0\|^2 \quad (3.14)$$

From [113], the above equation can be expanded as:

$$\psi = \phi(\mathbf{x}, \mathbf{x}) - \frac{2}{K} \sum_{i=1}^K \phi(\mathbf{x}, \mathbf{m}_i) + \frac{1}{K^2} \sum_{i,j=1}^K \phi(\mathbf{m}_i, \mathbf{m}_j) \quad (3.15)$$

Since Gaussian kernel function is employed, the first term and the last term in equation (3.15) are constants and could be discarded. The spherical potential (ψ) is defined as the value in squared distance mapping:

$$\psi = \frac{1}{K} \sum_{i=1}^K \phi(\mathbf{x}, \mathbf{m}_i) \quad (3.16)$$

The Gaussian kernel in equation (3.16) measures the similarity of the sample over existing rules in the network and can be expressed by the average of upper and lower firing strength. It is a direct measure of present knowledge in the sample [56, 57, 67, 92] and is given as:

$$\psi(t) = \sum_{i=1}^K \frac{f_i^{lo}(t) + f_i^{up}(t)}{2K} \quad (3.17)$$

A higher value of spherical potential shows the short distance of the current sample from contributing rules in the hyper-dimensional feature space, resulting in the similarity of knowledge in the network. A lower spherical potential indicates that the knowledge in the sample is novel. In the next section, the learning strategies of the system are described in detail.

3.2.3.1 Sample Delete Strategy

A sample is deleted without being learnt if its prediction error is less than delete threshold, E_d . Delete threshold prevents the network to learn similar samples, thereby avoiding over-training and reducing the computational effort. It is chosen according to the desired prediction accuracy. A higher value of delete threshold results in deletion of more samples without being learnt. The *sample-delete-strategy* helps the network avoid over-training by deleting samples with low information. Hence, E_d is chosen in the range [0.0001, 0.001] using grid search

FIGURE 3.3: McIT2FIS-GD learning mechanism

with five-fold cross-validation. The criterion is given as:

$$E(t) < E_d \quad (3.18)$$

Because the network is capable of providing the prediction error within the confidence level denoted by the delete threshold (E_d) for the current sample, the knowledge contained in that sample is considered already present in the network. Hence the sample is discarded.

3.2.3.2 Sample Learn Strategy

A sample can be learnt either by adding a rule or updating the parameters based on it. These criteria are known as (Rule Adding Criterion) and (Rule Updating Criterion), respectively. The new rule is added into the system if the sample contains significant knowledge and the existing rules cannot cover new sample effectively. Thus, we will define the rule addition criterion and the rule updating criterion to check against the prediction error presented by the contributing rules as well as the novelty of the sample.

1. The Rule Adding Criterion:

A new rule is grown if for the current sample, the predicted output is different from the desired output to a certain confidence level and the sample contains significant knowledge. In other words, the prediction error for the sample is very high and the novelty criterion is satisfied. The rule growing criterion is given as:

$$E(t) > E_a \quad \text{AND} \quad \psi(t) < E_S \quad (3.19)$$

where, E_a is the adding threshold and E_S is the novelty threshold. A higher value of E_a and a low value of E_S shows resistance to rule addition. For

problems chosen in this work, E_a and E_S can be found using grid search with five fold cross-validation in the range $[0.10, 0.30]$ and $[0.01, 0.60]$ respectively. On addition of a new rule to the network, E_a is self-adapted as:

$$E_a := (1 - \delta)E_a + \delta E(t) \quad (3.20)$$

where, δ is the slope parameter which decides the rate of increase of E_a . For the problems in this chapter, δ is set close to 0. E_a first lets the network capture the global knowledge and later fine tune it. A higher value of add threshold will allow the network to add few rules which will affect generalization, whereas too low a value will result in large network size. The novelty threshold, E_S is employed to decide how far a newly presented sample should be to the existing rule centers, in order to be added as a new rule. If this threshold is chosen closer to one (i.e., current sample can be close to existing rules in the projection space), many rules will be added to the network which might affect its generalization ability. On the other hand, if this threshold is chosen closer to zero, very few rules will be added to the network, that might in turn affect the predictability of the network. For newly added rule $(K + 1) - th$, the left and right center of $j - th$ input feature are initialized as:

$$\begin{aligned} [m_1^{j,K+1}, m_2^{j,K+1}] &= [x_j(t) - 0.1, x_j(t) + 0.1] \\ j &= 1, 2, \dots, n \end{aligned} \quad (3.21)$$

The width initialization of the new rule is decided by computing the distance of the sample to the nearest rule and is given as:

$$\begin{aligned} [\sigma^{K+1}] &= \min_{\forall i} [\|x(t) - m_1^i\|, \|x(t) - m_2^i\|] \\ i &= 1, 2, \dots, K \end{aligned} \quad (3.22)$$

The consequent weights are assigned with an aim to minimize the prediction error on addition of the new rule is minimal. In particular, the network with $K + 1$ rules needs to provide the predicted output $\hat{y}(t)$ the same as the actual output $y(t)$. It is given as:

$$y(t) = \hat{y}(t) = f[x(t), \theta'] \quad (3.23)$$

where, θ' represents the new parameter vector of the network with $K + 1$ rules. The lower and upper weights are set to the same value while initialization. Derived from the equation (3.10-3.12), the weights associated with $o - th$ output feature are assigned as:

$$w_l^{K+1,o} = w_r^{K+1,o} = \left(y_o(t) - \frac{\Omega}{\sum_{i=1}^{K+1} (f_i^{lo}(t) + f_i^{up}(t))} \right) \times \frac{\sum_{i=1}^{K+1} (f_i^{lo}(t) + f_i^{up}(t))}{f_{K+1}^{lo}(1 - q_r^o + q_l^o) + 1 - q_r^o + q_l^o} \quad (3.24)$$

where,

$$\Omega = \sum_{i=1}^K w_l^{io} ((1 - q_l^o) f_i^{up}(t) + q_l^o f_i^{lo}(t)) + \sum_{i=1}^K w_r^{io} ((1 - q_r^o) f_i^{lo}(t) + q_r^o f_i^{up}(t)) \quad (3.25)$$

2. The Rule Updating Criterion:

The network parameters are updated if the prediction error of the current sample is higher than parameter update threshold. The rule update criterion is as follows:

$$E_u < E(t) < E_a \quad (3.26)$$

where, E_u denotes the self-adaptive parameter update threshold and decide if the parameters of the rules are updated using the gradient descent based

learning algorithm. E_u is adaptive by decreasing over time and given by:

$$E_u = \max(\beta, E_d) \quad (3.27)$$

where,

$$\beta = \frac{E_d - E_u}{\text{No. of reserve sample}} \times \text{epoch_count} + E_u \quad (3.28)$$

A lower value of this threshold results in more rules being updated, which might in turn affect the generalization ability of the network. On the other hand, a higher value of this threshold results in fewer rules being updated leading to the underperformance of the network. One needs to choose this parameter in the range of [0.04, 0.20]. The output weights, rule centers, widths and control parameters are updated using a gradient descent based algorithm. Gradient descent algorithm updates the parameters over a number of iterations to find the local minimum of the objective function. The formulas of updating the parameters in the network are given:

$$m_1^{ij}(t+1) = m_1^{ij}(t) - \eta \frac{\partial E(t)}{\partial m_1^{ij}} \quad (3.29)$$

$$m_2^{ij}(t+1) = m_2^{ij}(t) - \eta \frac{\partial E(t)}{\partial m_2^{ij}} \quad (3.30)$$

$$\sigma^i(t+1) = \sigma^i(t) - \eta \frac{\partial E(t)}{\partial \sigma^i} \quad (3.31)$$

$$w_l^{io}(t+1) = w_l^{io}(t) - \eta \frac{\partial E(t)}{\partial w_l^{io}} \quad (3.32)$$

$$w_r^{io}(t+1) = w_r^{io}(t) - \eta \frac{\partial E(t)}{\partial w_r^{io}} \quad (3.33)$$

where η is the learning rate. The time dependence t is dropped for the sake of convenience in the description. For clarification, we consider the single-output case. The objective is to minimize the error function defined as follow:

$$E = \frac{1}{2}(y - \hat{y})^2 \quad (3.34)$$

where, y and $\hat{y}$ are the desired output and predicted output respectively. The equations for updating w_l^i are as follows:

$$\frac{\partial E}{\partial w_l^i} = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial \hat{y}}{\partial w_l^i} \right) = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial (y_{lo} + y_{up})}{\partial w_l^i} \right) = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial y_{lo}}{\partial w_l^i} \right) \quad (3.35)$$

where,

$$\frac{\partial E}{\partial \hat{y}} = (y - \hat{y}); \quad \frac{\partial y_{lo}}{\partial w_l^i} = \frac{(1 - q_l)f_i^{lo} + q_l f_i^{up}}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.36)$$

Similarly, the updating equation for w_r^i is given:

$$\frac{\partial E}{\partial w_r^i} = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial \hat{y}}{\partial w_r^i} \right) = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial (y_{lo} + y_{up})}{\partial w_r^i} \right) = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial y_{up}}{\partial w_r^i} \right) \quad (3.37)$$

where,

$$\frac{\partial y_{up}}{\partial w_r^i} = \frac{(1 - q_r)f_i^{lo} + q_r f_i^{up}}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.38)$$

The equations for updating q_l are as follows:

$$\frac{\partial E}{\partial q_l} = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial \hat{y}}{\partial q_l} \right) = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial (y_{lo} + y_{up})}{\partial q_l} \right) = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial y_{lo}}{\partial q_l} \right) \quad (3.39)$$

where,

$$\frac{\partial y_{lo}}{\partial q_l} = \frac{\sum_{i=1}^K w_l^i (f_i^{lo} - f_i^{up})}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.40)$$

Similarly, the updating equation for q_r is given:

$$\frac{\partial E}{\partial q_r} = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial \hat{y}}{\partial q_r} \right) = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial (y_{lo} + y_{up})}{\partial q_r} \right) = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial y_{up}}{\partial q_r} \right) \quad (3.41)$$

where,

$$\frac{\partial y_{up}}{\partial q_r} = \frac{\sum_{i=1}^K w_r^i (f_i^{up} - f_i^{lo})}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.42)$$

The derivations of the premise parts are expressed by the following:

For the left means:

$$\frac{\partial E}{\partial m_1^{ij}} = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial \hat{y}}{\partial m_1^{ij}} \right) = (y - \hat{y}) \left(\frac{\partial y_{lo} + \partial y_{up}}{\partial m_1^{ij}} \right) \quad (3.43)$$

where,

$$\frac{\partial y_{lo}}{\partial m_1^{ij}} = \frac{\partial \left(\frac{(1-q_l) \sum_{i=1}^K f_i^{up} w_l^i + q_l \sum_{i=1}^K f_i^{lo} w_l^i}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \right)}{\partial m_1^{ij}} = \quad (3.44)$$

$$= \frac{(1 - q_l) w_l^i \frac{\partial f_i^{up}}{\partial m_1^{ij}} + q_l w_l^i \frac{\partial f_i^{lo}}{\partial m_1^{ij}} - \left(\frac{\partial f_i^{up}}{\partial m_1^{ij}} + \frac{\partial f_i^{lo}}{\partial m_1^{ij}} \right) y_{lo}}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.45)$$

and

$$\frac{\partial y_{up}}{\partial m_1^{ij}} = \frac{\partial \left(\frac{(1-q_r) \sum_{i=1}^K f_i^{lo} w_r^i + q_r \sum_{i=1}^K f_i^{up} w_r^i}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \right)}{\partial m_1^{ij}} = \quad (3.46)$$

$$= \frac{(1 - q_r) w_r^i \frac{\partial f_i^{lo}}{\partial m_1^{ij}} + q_r w_r^i \frac{\partial f_i^{up}}{\partial m_1^{ij}} - \left(\frac{\partial f_i^{up}}{\partial m_1^{ij}} + \frac{\partial f_i^{lo}}{\partial m_1^{ij}} \right) y_{up}}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.47)$$

For the right means:

$$\frac{\partial E}{\partial m_2^{ij}} = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial \hat{y}}{\partial m_2^{ij}} \right) = (y - \hat{y}) \left(\frac{\partial y_{lo} + \partial y_{up}}{\partial m_2^{ij}} \right) \quad (3.48)$$

$$\frac{\partial y_{lo}}{\partial m_2^{ij}} = \frac{\partial \left(\frac{(1-q_l) \sum_{i=1}^K f_i^{up} w_l^i + q_l \sum_{i=1}^K f_i^{lo} w_l^i}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \right)}{\partial m_2^{ij}} = \quad (3.49)$$

$$= \frac{(1 - q_l) w_l^i \frac{\partial f_i^{up}}{\partial m_2^{ij}} + q_l w_l^i \frac{\partial f_i^{lo}}{\partial m_2^{ij}} - \left(\frac{\partial f_i^{up}}{\partial m_2^{ij}} + \frac{\partial f_i^{lo}}{\partial m_2^{ij}} \right) y_{lo}}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.50)$$

and

$$\frac{\partial y_{up}}{\partial m_2^{ij}} = \frac{\partial \left(\frac{(1-q_r) \sum_{i=1}^K f_i^{lo} w_r^i + q_r \sum_{i=1}^K f_i^{up} w_r^i}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \right)}{\partial m_2^{ij}} = \quad (3.51)$$

$$= \frac{(1-q_r) w_r^i \frac{\partial f_i^{lo}}{\partial m_2^{ij}} + q_r w_r^i \frac{\partial f_i^{up}}{\partial m_2^{ij}} - \left(\frac{\partial f_i^{up}}{\partial m_2^{ij}} + \frac{\partial f_i^{lo}}{\partial m_2^{ij}} \right) y_{up}}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.52)$$

For the width:

$$\frac{\partial E}{\partial \sigma^i} = \frac{\partial E}{\partial \hat{y}} \left(\frac{\partial \hat{y}}{\partial \sigma^i} \right) = (y - \hat{y}) \left(\frac{\partial y_{lo} + \partial y_{up}}{\partial \sigma^i} \right) \quad (3.53)$$

$$\frac{\partial y_{lo}}{\partial \sigma^i} = \frac{\partial \left(\frac{(1-q_l) \sum_{i=1}^K f_i^{up} w_l^i + q_l \sum_{i=1}^K f_i^{lo} w_l^i}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \right)}{\partial \sigma^i} = \quad (3.54)$$

$$= \frac{(1-q_l) w_l^i \frac{\partial f_i^{up}}{\partial \sigma^i} + q_l w_l^i \frac{\partial f_i^{lo}}{\partial \sigma^i} - \left(\frac{\partial f_i^{up}}{\partial \sigma^i} + \frac{\partial f_i^{lo}}{\partial \sigma^i} \right) y_{lo}}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.55)$$

$$\frac{\partial y_{up}}{\partial \sigma^i} = \frac{\partial \left(\frac{(1-q_r) \sum_{i=1}^K f_i^{lo} w_r^i + q_r \sum_{i=1}^K f_i^{up} w_r^i}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \right)}{\partial \sigma^i} = \quad (3.56)$$

$$= \frac{(1-q_r) w_r^i \frac{\partial f_i^{lo}}{\partial \sigma^i} + q_r w_r^i \frac{\partial f_i^{up}}{\partial \sigma^i} - \left(\frac{\partial f_i^{up}}{\partial \sigma^i} + \frac{\partial f_i^{lo}}{\partial \sigma^i} \right) y_{up}}{\sum_{i=1}^K (f_i^{lo} + f_i^{up})} \quad (3.57)$$

where,

$$\frac{\partial f_i^{up}}{\partial m_1^{ij}} = \frac{\partial f_i^{up}}{\partial \mu_{ij}^{up}} \frac{\partial \mu_{ij}^{up}}{\partial m_1^{ij}} = \begin{cases} f_i^{up} \frac{x_j - m_1^{ij}}{(\sigma^i)^2} & x_j \leq m_1^{ij} \\ 0 & otherwise \end{cases} \quad (3.58)$$

$$\frac{\partial f_i^{lo}}{\partial m_1^{ij}} = \frac{\partial f_i^{lo}}{\partial \mu_{ij}^{lo}} \frac{\partial \mu_{ij}^{lo}}{\partial m_1^{ij}} = \begin{cases} f_i^{lo} \frac{x_j - m_1^{ij}}{(\sigma^i)^2} & x_j > \frac{m_1^{ij} + m_2^{ij}}{2} \\ 0 & otherwise \end{cases} \quad (3.59)$$

$$\frac{\partial f_i^{up}}{\partial m_2^{ij}} = \frac{\partial f_i^{up}}{\partial \mu_{ij}^{up}} \frac{\partial \mu_{ij}^{up}}{\partial m_2^{ij}} = \begin{cases} f_i^{up} \frac{x_j - m_2^{ij}}{(\sigma^i)^2} & x_j > m_2^{ij} \\ 0 & otherwise \end{cases} \quad (3.60)$$

$$\frac{\partial f_i^{lo}}{\partial m_2^{ij}} = \frac{\partial f_i^{lo}}{\partial \mu_{ij}^{lo}} \frac{\partial \mu_{ij}^{lo}}{\partial m_2^{ij}} = \begin{cases} f_i^{lo} \frac{x_j - m_2^{ij}}{(\sigma^i)^2} & x_j \leq \frac{m_1^{ij} + m_2^{ij}}{2} \\ 0 & otherwise \end{cases} \quad (3.61)$$

$$\frac{\partial f_i^{up}}{\partial \sigma^i} = \frac{\partial (\prod_{j=1}^n \mu_{ij}^{up})}{\partial \sigma^i} = \sum_{j=1}^n \begin{cases} f_i^{up} \frac{(x_j - m_1^{ij})^2}{(\sigma^i)^3} & x_j < m_1^{ij} \\ f_i^{up} \frac{(x_j - m_2^{ij})^2}{(\sigma^i)^3} & x_j > m_2^{ij} \\ 0 & otherwise \end{cases} \quad (3.62)$$

$$\frac{\partial f_i^{lo}}{\partial \sigma^i} = \frac{\partial (\prod_{j=1}^n \mu_{ij}^{lo})}{\partial \sigma^i} = \sum_{j=1}^n \begin{cases} f_i^{lo} \frac{(x_j - m_2^{ij})^2}{(\sigma^i)^3} & x_j \leq \frac{m_1^{ij} + m_2^{ij}}{2} \\ f_i^{lo} \frac{(x_j - m_1^{ij})^2}{(\sigma^i)^3} & x_j > \frac{m_1^{ij} + m_2^{ij}}{2} \end{cases} \quad (3.63)$$

The learning rate must be chosen with care because too high learning rate will make the problem sometimes unable to converge while too low learning rate may lead to slow learning convergence.

3.2.3.3 Sample Reserve Strategy

When the sample is neither deleted or used for learning, it is reserved in the training data set for possible utilization in the subsequent epochs of learning. These three strategies are repeated for each of the incoming samples. It helps the network in efficient generalization. Next, we evaluate the performance of McIT2FIS-GD on a set of benchmark problems followed by a real-world wind speed prediction problem.

3.3 Performance Evaluation

In this section, the performance of McIT2FIS-GD is evaluated on a set of benchmark and a real world regression problem. First, the performance of McIT2FIS-GD is evaluated on non-linear system identification problem 1 [72] followed by system identification problem 2 [73]. Next, its performance is evaluated on MackeyGlass chaotic time series prediction problem [110] and a real world wind prediction problem [90]. The performance is compared against various type-1 neuro-fuzzy inference systems including eTS [66], Simple_eTS [111], SONFIN [72], SAFIS [73], support vector regression (SVR) and OS-fuzzy-ELM [80] and type-2 neuro-fuzzy inference systems including eT2FIS [19], RIT2NFS-WB [97], SEIT2FNN [20] and [18]. All the results except for McIT2FIS-GD have been adopted from literature.

3.3.1 Performance Measures

In this study, root mean squared error is employed to evaluate the performance of the systems. The Mackey-Glass time series prediction problem also employs non-destructive error index [67]. Root mean square error (RMSE) which measures the difference between the actual and the predicted output is used to measure the performance of algorithm on dynamical system identification problems. RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t))^2} \quad (3.64)$$

where, N indicates the total number of samples. Non-destructive error index (NDEI) [67] is also employed for Mackey-Glass time series forecasting problem and is defined as $RMSE$ divided by the standard deviation of the target. It is

given by:

$$NDEI = \frac{RMSE}{Standard\ Deviation} \quad (3.65)$$

3.3.2 Identification of a Nonlinear System 1

The performance of McIT2FIS-GD is evaluated on nonlinear system identification problem 1 [72], which is given as:

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \quad (3.66)$$

where, the one step ahead output of the system, $y(t+1)$, depends nonlinearly on its current output, $y(t)$, and an input, $u(t) = \sin(2\pi t/100)$. The training dataset consists of 50,000 samples and testing dataset consists of 200 samples.

TABLE 3.1: Performance comparison of McIT2FIS-GD on non-linear system identification problem 1

	Algorithm	Rules	Training RMSE	Testing RMSE
Type-1	eTS	19	-	0.008
	SONFIN	10	-	0.013
	SAFIS	8	-	0.011
Type-2	SIT2FNN	3	-	0.024
	eT2FIS	14	-	0.053
	McIT2FIS-GD	5	0.008	0.008

McIT2FIS-GD has been compared with other type-1 and type-2 algorithms available in the literature. Its performance is compared with SONFIN, eTS and SAFIS in type-1 literature and SIT2FNN and eT2FIS in interval type-2 literature. The results of compared type-1 systems have been adopted from [72] and interval type-2 systems from [18]. Table 3.1 shows the number of rules, train and test RMSE for

the considered algorithms. It can be observed that the performance of McIT2FIS-GD is better than the compared algorithms. It achieves an error of one order less than SIT2FNN using a similar number of rules. In case of eT2FIS, McIT2FIS-GD attains better performance by employing one-third of the total number of rules used by eT2FIS. The experiment on the number of samples deleted and the number of samples used for update in each iteration of the learning algorithm has been conducted in author's paper [56]. It has been stated that in the initial iterations, maximum number of samples are used for update while very few samples are deleted, however, at a later stage, maximum number of samples are deleted and a few samples are used for updating the knowledge. Next, the performance of McIT2FIS-GD is evaluated on non-linear system identification problem 2 [73].

3.3.3 Identification of a Nonlinear System 2

The performance of McIT2FIS-GD is evaluated on nonlinear system identification problem 2, as given in [73]. It is defined as:

$$y(t) = \frac{y(t-1) * y(t-2) * (y(t-1) - 0.5)}{1 + y^2(t-1) + y^2(t-2)} + u(t-1) \quad (3.67)$$

where, $u(t)$ and $y(t)$ are the input and output at the t^{th} instant. The input to the system is given as $u(t) = \sin(2t/25)$. For the purpose of training and testing 5000 and 200 observation data are produced.

The performance of McIT2FIS-GD is compared with eTS, Simpl_eTS and SAFIS in Table. Table 3.2 shows the number of rules, Train and Test RMSE. The result of all the algorithms except McIT2FIS-GD is adopted from [73]. From Table 3.2, it may be seen that McIT2FIS-GD performs better than all the compared algorithms using a lesser rule count. The logarithmic prediction error of the system during training process is demonstrated in Figure 3.4. It can be noticed from the

figure that the prediction error decreases significantly within the first four hundred epochs. This can be explained by the fact that gradient descent algorithm optimizes the network parameters based on the prediction error alone.

FIGURE 3.4: Training logarithmic prediction error for non-linear system identification problem 2 using McIT2FIS-GD

The Figure 3.5 shows the plot for actual and predicted output for testing data set. It can be seen from the figure that the training data has been accurately generalized. Next, we evaluate the performance of McIT2FIS-GD on the MackeyGlass problem [110].

TABLE 3.2: Performance comparison of McIT2FIS-GD on non-linear system identification problem 2

Algorithm	Rules	Training RMSE	Testing RMSE
eTS	49	0.029	0.021
SAFIS	17	0.054	0.022
Simpl.eTS	22	0.052	0.025
McIT2FIS-GD	6	0.021	0.015

FIGURE 3.5: Actual vs predicted output for non-linear system identification 2 using McIT2FIS-GD

3.3.4 Mackey-Glass Time Series Problem

The Mackey-Glass chaotic time series problem [110] is given as:

$$\frac{\partial x}{\partial t} = \frac{0.2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0.1x(t) \quad (3.68)$$

The aim of the problem is to predict the future values $x(t+85)$ of the system based on current and past values, $[x(t - 18), x(t - 12), x(t - 6), x(t)]$. The parameters for this problem are set as: $\tau = 17$ and $x(0) = 1.2$. A total of three thousand five hundred samples were generated, of which three thousand were used for training and five hundred were used for testing.

The performance of McIT2FIS-GD is compared with SAFIS, Simple_eTS and eTS in type-1 literature, SEIT2FNN and RIT2NFS-WB in type-2 literature in Table 3.3. The results of type-1 systems have been adopted from [73] and type-2 systems from [97]. It can be seen that McIT2FIS-GD performs better than compared algorithms by using lesser number of rules. Testing NDEI of the proposed algorithm is competitive in comparison with other methods, which shows effective generalization of the network.

TABLE 3.3: Performance comparison of McIT2FIS-GD on Mackey-Glass time series problem-85

Type	Algorithm	Rules	Testing NDEI
Type-1	eTS	9	0.380
	Simp_eTS	11	0.394
	SAFIS	6	0.376
Type-2	SEIT2FNN	13	0.206
	RIT2NFS-WB	13	0.208
	McIT2FIS-GD	5	0.043

3.3.5 Wind Speed Prediction Problem

Recently, wind energy has emerged as a perennial source of energy due to a need for a clean energy source. The power generated by the wind turbines is non-schedulable in nature due to changing weather conditions [114]. Hence, there is a need to predict wind speed for efficient wind power generation. Since the prediction problem can be casted as a function approximation problem, McIT2FIS-GD is employed for the wind prediction problem. The real-world wind data set is obtained from Iowa (USA) Department of Transport at the location, Washington. The data sampled for every 10 minutes is downloaded between over a period of February 1, 2011 to February 28, 2011. The data is averaged hourly from which ten features (speed and direction of past five hours) are extracted. The training data set consists of five hundred samples and the testing dataset consists of one hundred samples.

Figure 3.6 and 3.7 show the actual output versus predicted output for training data and testing dataset respectively. It can be observed from the figure that McIT2FIS-GD is able to generalize well the underlying functional relationship between the past and future wind speed. The performance of McIT2FIS-GD is compared with SVR, OS-fuzzy-ELM and functional link artificial neural network (FLANN) [115] in type-1 literature and SIT2FNN in type-2 literature in Table 3.4. Table 3.4

illustrates the number of rules employed, train and test root mean square error for all the compared algorithms. It may be seen from the table that McIT2FIS-GD performs better than the compared algorithms, which proves that the use of meta-cognition has helped McIT2FIS-GD obtain better accuracy.

FIGURE 3.6: Actual and predicted wind speed for training data using McIT2FIS-GD

TABLE 3.4: Performance comparison for wind prediction problem

	Algorithm	Rules	Training RMSE	Testing RMSE
Type-1	SVR	-	0.197	0.214
	OS-fuzzy-ELM	-	0.255	0.286
	FLANN	30	-	0.532
Type-2	SIT2FNN	4	0.194	0.187
	McIT2FIS-GD	4	0.136	0.132

To further discuss the contribution of meta-cognitive learning mechanism in this research, two separate experiments are conducted to compare the performance

FIGURE 3.7: Actual and predicted wind speed for testing data using McIT2FIS-GD

of a plain interval type-2 neuro-fuzzy system (IT2FIS) with an integrated meta-cognitive system. In the first experiment, the rule growing algorithm is validated by training two systems under a maximum number of twenty rules. In the second experiment, IT2FIS is trained using the rules and samples selected by McIT2FIS-GD to address the wind speed prediction problem. The results are available in Table 3.5. It can be observed from the table that IT2FIS employs the maximum number of rules to approximate the underlying function, which is not only redundant but also increases the computational time of the training phase. On the other hand, the system integrated meta-cognition is capable of controlling rule growth to obtain better training and testing accuracy. It should be noted that the computation time is reduced significantly. The second experiment shows that by exploiting the knowledge captured by meta-cognitive component, the performance of IT2FIS is clearly better. The reduction in root mean square error and CPU time have indicated the advantages of the learning strategies in the system.

TABLE 3.5: Experiments on meta-cognitive learning mechanism for wind speed prediction problem

	Algorithm	Rules	Training RMSE	Testing RMSE	CPU time (s)
Experiment I	IT2FIS	20	0.229	0.234	62.1
	McIT2FIS-GD	4	0.136	0.132	4.3
Experiment II	IT2FIS	4	0.179	0.219	20.3

3.4 Summary

An evolving interval type-2 neuro-fuzzy inference system and its meta-cognitive learning mechanism have been proposed in this chapter. The system is capable of evolving its structure and parameters learning is based on a gradient descent algorithm. Comparative results have indicated advantages of the proposed algorithm over others state-of-the-art type-1 and type-2 neuro-fuzzy inference systems in the literature. In the next chapter, the system is developed to handle sequentially arriving data. The updating algorithm is based on an extended Kalman filter method.

Chapter 4

A Sequential Meta-cognitive Learning Algorithm for Interval Type-2 Fuzzy Inference System

In the previous chapter, an evolving neuro-fuzzy inference system has been proposed. The system employing a fast inference mechanism is capable of solving function approximation problems while the network parameters are updated based on a gradient descent algorithm. In this chapter, a fully sequential learning algorithm is developed for the proposed interval type-2 neuro-fuzzy inference system. The extended Kalman filtering-based learning scenario is employed for parameters learning. In order to evaluate the performance of the proposed algorithm, benchmark approximation problems are considered. Competitive results indicate the advantages of the algorithm.

4.1 Introduction

The previous chapter has introduced an evolving interval type-2 fuzzy inference system and its meta-cognitive learning algorithm. The system is able to grow its network structure by monitoring the knowledge present in the sample to cope with rapidly changing learning environment. The learning algorithm relies on an iterative learning mechanism and selects learning strategy based on an inspired human meta-cognition. The system has overcome the limitations of conventional neuro-fuzzy inference systems employing static network architecture.

In case of various practical problems, time-varying data arrives in sequence and an online learning mechanism affects the performance of the neuro-fuzzy inference system significantly [18–20, 56, 64, 67, 73]. In addition, the initialization of appropriate upper and lower centers, as well as the membership function width of a new fuzzy rule in the network, helps the system to generalize more accurately. From these observations, in this chapter, we have developed McIT2FIS-GD as a fully sequential learning algorithm. In order to address the above-mentioned problems, an extended Kalman filtering-based approach has been employed to adapt the parameters of the network. In the learning process, the samples are presented sequentially to the system and learnt only once. The system initializes a new rule by controlling the expansion of fuzzy rule centers and the overlap between newly added rule and the existing rules in the network.

The proposed meta-cognitive sequential learning algorithm for interval type-2 neuro-fuzzy inference is indicated as meta-cognitive interval type-2 fuzzy inference system EKF-based, or McIT2FIS-EKF. The learning mechanism of the algorithm is formulated on a five-layer network realizing Takagi-Sugeno-Kang fuzzy inference. The input layer passes data towards the membership layer to obtain Gaussian membership values. The firing layer calculates the strength of rules before employing interval reduction to approximate the output. The controlling q factors [88] together with upper and lower weights enable the construction of two bounds in the output layer. Meta-cognition which facilitates regulation

in learning process [67, 116] helps in approximating functional relationship between input-output and growing the architecture effectively [21, 57, 67, 92, 117]. The algorithm in this chapter employs a modified meta-cognitive learning strategies to determine the structure of the network. McIT2FIS-EKF learning starts with zero rules and evolves the architecture automatically. Based on the current sample and knowledge contained in the network, the learning algorithm selects a suitable learning strategy ('sample-delete' or 'sample-learn' or 'sample-reserve'). In *sample-learn-strategy*, the learning algorithm adds a new rule or updates the existing parameters using an extended Kalman filter approach. In *sample-delete-strategy*, the sample is discarded without being used in the learning process. In *sample-reserve-strategy*, the sample is kept at the end of data stream for future use.

The performance of the proposed algorithm is assessed based on a benchmark non-linear system identification problem and Mackey-Glass time series prediction problem stated in the literature [72, 110]. The performance is compared with other state-of-the art algorithms including evolving Takagi-Sugeno model (eTS)[66], sequential adaptive fuzzy inference system (SAFIS) [73], meta-cognitive neuro-fuzzy inference system (McFIS)[67], parsimonious network based on fuzzy inference system (PANFIS) [94], generic evolving neuro-fuzzy inference system (GENEFIS) [95], a self-evolving interval type-2 fuzzy neural network (SEIT2FNN) [20], simplified interval type-2 fuzzy neural network (SIT2FNN) [18] and meta-cognitive interval type-2 fuzzy inference system gradient descent (McIT2FIS-GD) [56]. The results indicate that McIT2FIS-EKF clearly shows advantages over other systems in the literature.

4.2 Problem Definition

The leaning algorithm adapts the system by processing training samples only once. Each training sample is denoted as a pair $(\mathbf{x}(t), \mathbf{y}(t))$, where $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)] \in R^{1 \times n}$ is *n-dimensional* input vector and the output $\mathbf{y}(t) =$

$[y_1(t), y_2(t), \dots, y_M(t)] \in R^{1 \times M}$ is M -dimensional vector. The objective of the algorithm is to estimate the functional relationship $f[\cdot]$ between the input and the output ($x \rightarrow y$) so that the predicted output:

$$\hat{y} = f[x(t), \theta] \quad (4.1)$$

is an accurate approximation of the actual output. It is noted that vector θ represents the parameters of the rules. The error for t -th sample measures the difference between the predicted output and the desired output and is given as:

$$\begin{aligned} e(t) &= [e_1(t), e_2(t), \dots, e_M(t)] \\ e_o(t) &= y_o(t) - \hat{y}_o(t) \quad o = 1, 2, \dots, M \end{aligned} \quad (4.2)$$

The root mean square prediction error for t -th sample is as follow:

$$E(t) = \sqrt{\frac{1}{M} \sum_{o=1}^M (e_o(t))^2} \quad (4.3)$$

4.3 Sequential Meta-cognitive Interval Type-2 Fuzzy Inference System Architecture

This section describes the architecture of the proposed fuzzy neural network. The objective is to approximate functional relationship between the input x and the output y . The architecture is based on an interval type-2 neuro-fuzzy inference system. The premise part of a rule employs interval-type 2 fuzzy membership functions with uncertain means and fixed standard deviation. The structure consists of five-layer network realizing Takagi-Sugeno-Kang fuzzy inference mechanism and is depicted in Figure 4.1. Assuming that the system learns n input features and has grown K rules after $t - 1$ samples. The detailed inference of each layer at the $t - th$ sample is presented as follows:

FIGURE 4.1: Architecture of the interval type-2 neuro fuzzy system.

Layer 1- Input layer: This layer contains n nodes representing n number of input features. The input is passed directly to layer 2, membership layer. The output of j -th node is given:

$$u_j(t) = x_j(t); j = 1, 2, \dots, n \quad (4.4)$$

Layer 2- Membership layer: This layer calculates the upper and the lower membership strength of each j -th feature in every i -th rule. The employed interval type-2 fuzzy membership function with uncertain means and fixed standard deviation is demonstrated in Figure 4.2. The formulas are given by:

$$\mu_{ij}^{up}(t) = \begin{cases} \phi(m_1^{ij}, \sigma^i, u_j(t)) & u_j(t) < m_1^{ij} \\ 1 & m_1^{ij} \leq u_j(t) \leq m_2^{ij} \\ \phi(m_2^{ij}, \sigma^i, u_j(t)) & u_j(t) > m_2^{ij} \end{cases} \quad (4.5)$$

$$\mu_{ij}^{lo}(t) = \begin{cases} \phi(m_2^{ij}, \sigma^i, u_j(t)) & u_j(t) \leq \frac{m_1^{ij} + m_2^{ij}}{2} \\ \phi(m_1^{ij}, \sigma^i, u_j(t)) & u_j(t) > \frac{m_1^{ij} + m_2^{ij}}{2} \end{cases} \quad (4.6)$$

where,

$$\phi(m^{ij}, \sigma^i, u_j) = \exp\left(-\frac{(u_j - m^{ij})^2}{2(\sigma^i)^2}\right); i = 1, 2, \dots, K \quad (4.7)$$

where m_1, m_2 are the left and right center of the interval type-2 Gaussian membership function accordingly, and σ^i is the width of the i -th rule. The output of this layer can be represented by the intervals $[\mu_{ij}^{lo}(t), \mu_{ij}^{up}(t)]$.

FIGURE 4.2: Footprint of uncertainty using uncertain means and fixed standard deviation.

Layer 3- Firing layer: Each node in this layer which represents the upper and lower firing strength of a rule is calculated by the algebraic product of membership values of a rule. The output of this layer is given as:

$$[f_i^{lo}(t), f_i^{up}(t)]; i = 1, 2, \dots, K \quad (4.8)$$

where,

$$f_i^{lo}(t) = \prod_{j=1}^n \mu_{ij}^{lo}(t) \quad (4.9)$$

and

$$f_i^{up}(t) = \prod_{j=1}^n \mu_{ij}^{up}(t) \quad (4.10)$$

Layer 4- Output processing layer: This layer adjusts the proportion of the upper and lower bounds and passes them to the output layer. Instead of using Karnick-Mendel iterative procedure [107] to find lower and upper endpoints, the control parameters q_l and q_r [18] are employed to increase the learning speed and minimize the computational complexity. The output of the firing layer is combined with consequent weight parameters, and the contributions of rules are adjusted to process the output. The two bounds of o -th output feature at time t -th are $y_{lo}^o(t)$ and $y_{up}^o(t)$. The formulas to calculate y_{lo} and y_{up} in this layer are given by:

$$y_{lo}^o(t) = \frac{(1 - q_l^o) \sum_{i=1}^K f_i^{up}(t) w_l^{io} + q_l^o \sum_{i=1}^K f_i^{lo}(t) w_l^{io}}{\sum_{i=1}^K (f_i^{lo}(t) + f_i^{up}(t))}, \quad (4.11)$$

$o = 1, 2, \dots, M$

$$y_{up}^o(t) = \frac{(1 - q_r^o) \sum_{i=1}^K f_i^{lo}(t) w_r^{io} + q_r^o \sum_{i=1}^K f_i^{up}(t) w_r^{io}}{\sum_{i=1}^K (f_i^{lo}(t) + f_i^{up}(t))}, \quad (4.12)$$

$o = 1, 2, \dots, M$

Layer 5- Output layer: The output of this layer is the combination of y_{lo} and y_{up} from output processing layer. It is given by:

$$\hat{y}_o(t) = \frac{1}{2} y_{lo}^o(t) + y_{up}^o(t); \quad o = 1, 2, \dots, M \quad (4.13)$$

The above layers create a computationally fast inference and are able to evolve in the learning process. Next section describes the sequential learning algorithm for the proposed system.

4.4 Meta-cognitive Learning Algorithm based Extended Kalman Filtering

During the sequential learning process, a sample is presented one by one to the network. The meta-cognitive learning mechanism [56, 67, 92] monitors the contained knowledge of the sample to make decision on appropriate learning strategy. The three strategies are *sample-delete-strategy* (delete the sample without learning), *sample-learn-strategy* (learn the knowledge in the sample) and *sample-reserve-strategy* (reserve the sample for future learning). In order to measure the novel knowledge in the current sample, prediction error and spherical potential are calculated. The two indicators are then monitored by meta-cognitive learning mechanism, and the learning is controlled in a sequential process. The prediction error $E(t)$ for the sample t -th is given in equation (4.3). In order to effectively measure the novelty in data, spherical potential [67] based on the distance between the sample and significantly contributing rules in the hyper-dimensional feature space is employed. Spherical potential in this algorithm is given by:

$$\psi(t) = \sum_{i=1}^{K_c} \frac{f_i^{lo}(t) + f_i^{up}(t)}{2K_c} \quad (4.14)$$

where K_c is the number of significant rules in the network. A rule i -th is considered significantly contributed if its firing strength $F_i(t) = (f_i^{lo}(t) + f_i^{up}(t))/2 > 0.1$. By employing significant rules to calculate spherical potential, the novelty criterion is assessed more strictly and the impact of less significantly contributed rules is eliminated. From the definition of $\psi(t)$, it is indicated that a smaller spherical potential (close to zero) means novel knowledge and a higher spherical potential (close to one) means that the knowledge in the sample has been acquired

in the network. The following subsections shall specify the meta-cognitive learning algorithm of the system.

4.4.1 Sample Delete Strategy

The sample is deleted if its prediction error is smaller than the delete threshold, E_d . E_d is set close to zero. For the problems in this chapter, E_d is chosen in the range $[0.0001, 0.001]$. The condition is given as:

$$E(t) < E_d \quad (4.15)$$

4.4.2 Sample Learn Strategy

New rule is added into the system if knowledge in the current sample is novel to the contributing rules. In this case, a new rule is grown in the system by adding new nodes into the network: n nodes in the membership layer representing n input features, two nodes in the firing layer representing upper and lower firing strength. The network then consists of $K + 1$ rules. Alternatively, the current sample could be learnt to update the parameters of existing rules in the network. In this strategy, we define the rule growing criterion or rule updating criterion by monitoring the prediction error and the novelty of the sample when it is presented to the network.

4.4.2.1 The Rule Adding Criterion

If the prediction error of the current sample is higher than a threshold and the distance of the sample to contributing rules in the network is lower than a certain level, the knowledge in the sample is decided as novel and a new rule is added into the network. In order to control the growing process, we also check upon the maximum number of existing rules. The criterion for rule adding is as follow:

$$E(t) > E_a \quad \text{AND} \quad \psi(t) < E_S \quad \text{AND} \quad K < N_{max} \quad (4.16)$$

where, E_a is the adding threshold, E_S is the novelty threshold and N_{max} is the maximum number of rules to be added. The two thresholds are self-adaptive and they are updated after a new rule is added into the network. γ_1 and γ_2 are the slopes at which E_a increases and E_S decreases in updating formulas below:

$$E_a := (1 - \gamma_1)E_a + \gamma_1 E(t) \quad (4.17)$$

$$E_S := (1 - \gamma_2)E_S - \gamma_2 \psi(t) \quad (4.18)$$

The aim of self-adaptive controlling parameters in this algorithm is to let the system capture the general knowledge at the beginning stage of the learning process and later fine-tune it. The two parameters γ_1 and γ_2 are set close to zero. γ_2 is determined based on the number of samples in the training dataset. The new $(K + 1) - th$ rule centers are initialized as follows:

$$\begin{aligned} [m_1^{j,K+1}, m_2^{j,K+1}] &= [x_j(t) - \xi, x_j(t) + \xi] \\ j &= 1, 2, \dots, n \end{aligned} \quad (4.19)$$

where, ξ is the controlling extension parameter of the new rule. ξ is chosen in the range $[0.05, 0.25]$. On one hand, the lower value of ξ makes the new rule cover fewer samples in the data set which results in insufficient generalization. On the other hand, a new rule with too high ξ is capable of generalizing more samples, which shall affect the accuracy of the system. The width of the new rule is initialized as:

$$[\sigma^{K+1}] = \kappa \times \min_{\forall i} [\|x(t) - m_1^i\|, \|x(t) - m_2^i\|] \quad (4.20)$$

$$i = 1, 2, \dots, K$$

where, κ controls the overlap of the newly added rule and existing rules in the network. In this thesis, κ lies in the range [0.4, 0.7] to ensure a positive effect on the performance of the system. It has been shown that the initialization of new output weights should completely exploit the localization property of Gaussian membership function [118]. The new output weights w_l^{K+1} and w_r^{K+1} are initialized equally and the initialization aims to minimize the prediction error with the contribution of new rule. After adding a new rule, the network consisting of $K+1$ rules provides the predicted output $\hat{y}(t)$, which should be equal to the actual output $y(t)$. The formula to initialize new weight is as follow:

$$w_l^{K+1} = w_r^{K+1} = \frac{y(t)(2K\psi(t) + 1 + f_{K+1}^{lo}) - \hat{y}(t)2K\psi(t)}{(1 - q_r + q_l)f_{K+1}^{lo} + 1 - q_l + q_r} \quad (4.21)$$

4.4.2.2 The Rule Updating Criterion

The parameters of the network are updated based on the knowledge in the current sample. If the prediction error is below the adding threshold and above an updating threshold, E_u , and the spherical potential of the sample is higher than the novelty threshold then the existing rules in the network are adapted. The detailed condition for this learning strategy is as follows:

$$E_u < E(t) < E_a \quad \text{AND} \quad \psi(t) > E_s \quad (4.22)$$

where, E_u is the updating threshold. E_u is self-adaptive and set in the range [0.01 0.1] for all the problems in this chapter. When the updating strategy is decided, the

parameter update threshold E_u and novelty threshold E_S are self-adapted based on the following equations:

$$E_u := (1 - \gamma_1)E(t) + \gamma_1 E_u \quad (4.23)$$

$$E_S := (1 - \gamma_2)E_S + \gamma_2 \psi(t) \quad (4.24)$$

The two slope control parameters γ_1 and γ_2 are used to allow the network to initially capture global knowledge and later to be tuned. The parameter vector $\boldsymbol{\theta} = [\mathbf{m}_1, \mathbf{m}_2, \boldsymbol{\sigma}, \mathbf{w}_l, \mathbf{w}_r, q_l, q_r]$ of the network is the z - *dimensional* vector of lower and upper rules center, rule width, output weights and two q control factors q_l, q_r . The dimension of the parameter vector depends on the number of rules grown in the network. Typically, it is given as: $z = K(2n + 3) + 2$. The parameter $\boldsymbol{\theta} \in R^{1 \times z}$ is updated as:

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \mathbf{e}(t)\mathbf{G}^T. \quad (4.25)$$

where, $\mathbf{e}(t)$ is the error defined in equation (4.2) and $\mathbf{G} \in R^{z \times M}$ is the Kalman gain matrix. It is given by:

$$\mathbf{G} = \mathbf{P}\mathbf{H}[\mathbf{R} + \mathbf{H}^T\mathbf{P}\mathbf{H}]^{-1} \quad (4.26)$$

where, $\mathbf{P} \in R^{z \times z}$ is the error covariance matrix of network parameters, $\mathbf{R} = r_0\mathbf{I}_{M \times M}$ is the variance of measurement noise and $\mathbf{H} \in R^{z \times M}$ is the gradient matrix of the predicted output with respect to network parameters. $\mathbf{P}$ is initialized as $\mathbf{P} = p_0\mathbf{I}_{z \times z}$ and is updated as:

$$\mathbf{P} = [\mathbf{I}_{z \times z} - \mathbf{G}\mathbf{H}^T]\mathbf{P} + q_0\mathbf{I}_{z \times z} \quad (4.27)$$

where, p_0 is initialized error covariance and is set greater than 1; q_0 is step size control parameter and is set close to 0. When a new rule is added into the network, the covariance matrix is updated by [73, 119]:

$$P = \begin{bmatrix} P & \mathbf{0} \\ \mathbf{0} & p_0 \mathbf{I}_{(2n+3) \times (2n+3)} \end{bmatrix} \quad (4.28)$$

It should be noted that $\mathbf{I}$ is the identity matrix. The following equation is the gradient matrix of predicted output with respect to network parameters.

$$\mathbf{H} = \begin{bmatrix} \frac{\partial \hat{y}_1}{\partial m_1^{11}} & \frac{\partial \hat{y}_2}{\partial m_1^{11}} & \cdots & \frac{\partial \hat{y}_o}{\partial m_1^{11}} & \cdots & \frac{\partial \hat{y}_M}{\partial m_1^{11}} \\ \vdots & \vdots & & \vdots & & \vdots \\ \frac{\partial \hat{y}_1}{\partial m_1^{nK}} & \frac{\partial \hat{y}_2}{\partial m_1^{nK}} & \cdots & \frac{\partial \hat{y}_o}{\partial m_1^{nK}} & \cdots & \frac{\partial \hat{y}_M}{\partial m_1^{nK}} \\ \frac{\partial \hat{y}_1}{\partial m_2^{11}} & \frac{\partial \hat{y}_2}{\partial m_2^{11}} & \cdots & \frac{\partial \hat{y}_o}{\partial m_2^{11}} & \cdots & \frac{\partial \hat{y}_M}{\partial m_2^{11}} \\ \vdots & \vdots & & \vdots & & \vdots \\ \frac{\partial \hat{y}_1}{\partial m_2^{nK}} & \frac{\partial \hat{y}_2}{\partial m_2^{nK}} & \cdots & \frac{\partial \hat{y}_o}{\partial m_2^{nK}} & \cdots & \frac{\partial \hat{y}_M}{\partial m_2^{nK}} \\ \frac{\partial \hat{y}_1}{\partial \sigma^1} & \frac{\partial \hat{y}_2}{\partial \sigma^1} & \cdots & \frac{\partial \hat{y}_o}{\partial \sigma^1} & \cdots & \frac{\partial \hat{y}_M}{\partial \sigma^1} \\ \vdots & \vdots & & \vdots & & \vdots \\ \frac{\partial \hat{y}_1}{\partial \sigma^K} & \frac{\partial \hat{y}_2}{\partial \sigma^K} & \cdots & \frac{\partial \hat{y}_o}{\partial \sigma^K} & \cdots & \frac{\partial \hat{y}_M}{\partial \sigma^K} \\ \frac{\partial \hat{y}_1}{\partial w_l^{11}} & \frac{\partial \hat{y}_2}{\partial w_l^{12}} & \cdots & \frac{\partial \hat{y}_o}{\partial w_l^{1\sigma}} & \cdots & \frac{\partial \hat{y}_M}{\partial w_l^{1M}} \\ \vdots & \vdots & & \vdots & & \vdots \\ \frac{\partial \hat{y}_1}{\partial w_l^{K1}} & \frac{\partial \hat{y}_2}{\partial w_l^{K2}} & \cdots & \frac{\partial \hat{y}_o}{\partial w_l^{K\sigma}} & \cdots & \frac{\partial \hat{y}_M}{\partial w_l^{KM}} \\ \frac{\partial \hat{y}_1}{\partial w_r^{11}} & \frac{\partial \hat{y}_2}{\partial w_r^{12}} & \cdots & \frac{\partial \hat{y}_o}{\partial w_r^{1\sigma}} & \cdots & \frac{\partial \hat{y}_M}{\partial w_r^{1M}} \\ \vdots & \vdots & & \vdots & & \vdots \\ \frac{\partial \hat{y}_1}{\partial w_r^{K1}} & \frac{\partial \hat{y}_2}{\partial w_r^{K2}} & \cdots & \frac{\partial \hat{y}_o}{\partial w_r^{K\sigma}} & \cdots & \frac{\partial \hat{y}_M}{\partial w_r^{KM}} \\ \frac{\partial \hat{y}_1}{\partial q_l^1} & \frac{\partial \hat{y}_2}{\partial q_l^2} & \cdots & \frac{\partial \hat{y}_o}{\partial q_l^p} & \cdots & \frac{\partial \hat{y}_M}{\partial q_l^M} \\ \frac{\partial \hat{y}_1}{\partial q_r^1} & \frac{\partial \hat{y}_2}{\partial q_r^2} & \cdots & \frac{\partial \hat{y}_o}{\partial q_r^p} & \cdots & \frac{\partial \hat{y}_M}{\partial q_r^M} \end{bmatrix} \quad (4.29)$$

The gradient of each column in the above equation can be adopted from the derivation described in detail from equation (3.35) to (3.63).

4.4.3 Sample Reserve Strategy

Similar to the *sample-reserve-strategy* in the previous chapter, the sample is pushed to the end of data stream for training in the later stage if it does not satisfy all the above conditions. In other words, the sample is reserved for later processing if its contained knowledge is not a priority to the system. The self-adaptive nature of the adding and updating threshold given in *sample-learn-strategy* facilitates the usage of these samples in later learning stages. The three learning strategies formulating the self-regulate meta-cognitive learning mechanism of the proposed IT2FIS are applied for every sample in the training process. The system is summarized in Algorithm 1. In the next section of this chapter, we discuss the performance of McIT2FIS-EKF on the problems of function approximation and compare it with the other algorithms in the literature.

4.5 Performance Evaluation

In the previous section, the detailed principles of the meta-cognitive sequential learning algorithm for IT2FIS have been introduced. In this section, the proposed algorithm is evaluated based on standard benchmark function approximation problems. Experiment on non-linear system identifications discussed in [72, 73] is first studied. Next, we conduct an experiment on a stochastic Mackey-Glass time series problem [110]. The performance of McIT2FIS-EKF is compared with other type-1 and type-2 fuzzy systems in the literature including SAFIS [73], eTS [66], McFIS [67], SIT2FNN [18], SEIT2FNN [20] and McIT2FIS-GD [56]. The results except for McIT2FIS-EKF are adopted from the literature.

The performance of the algorithm has been evaluated in a MATLAB R2013b environment on a Windows system with Xeon CPU and 16GB RAM. The comparative results show the strengths of meta-cognitive sequential learning algorithm over other neuro-fuzzy inference systems.

Algorithm 1: Pseudo-code for McIT2FIS-EKF

```
while samples in training data do
  for each input  $x(t)$  do
    Calculate the output (eqn. 4.4-4.13), prediction error (eqn. 4.3),
    spherical potential (eqn. 4.14)
    if  $E(t) < E_d$  then
      Delete sample from training sequence.
    else if  $E(t) > E_a$  AND  $\psi(t) < E_S$  AND  $K < N_{max}$  then
      Add a new rule to the network (eqn. 4.19-4.21)
      The self-adaptive adding threshold and novelty threshold
      are updated (eqn. 4.17-4.18)
    else if  $E_u < E(t) < E_a$  AND  $\psi(t) > E_S$  then
      Update parameters of existing rules (eqn. 4.25)
      The self-adaptive learning threshold and novelty threshold
      are updated (eqn. 4.23-4.24)
    else
      Push the sample to the end of the training pool.
      if a sample is not learnt after few processing
      then
        Remove it from data stream.
    end
  end
end
```

4.5.1 System Identification Problem 1

The one-time step ahead output of discrete-time nonlinear dynamical system [67, 72] mentioned in Section 3.4.2 is used to evaluate the performance of the algorithms. The aim of the problem is to predict current value $y(k+1)$ based on three past values. The sample distribution is the range of $[-1.5, 1.5]$. The training set has 50,000 samples and testing set has 200 samples. The root mean square error, computation time in second and number of rules for this problem are described in Table 4.1. It is observed that McIT2FIS-EKF is able to identify the output with minor errors and nearly the same number of rules. In order to examine the sequential learning algorithm, the percentage of samples required in training (PS) is used as an additional measure of comparison. A merely 85.09% of total data samples has been learnt while other type-1 such as eTS and SAFIS employ 100% of samples for training [66], [73]. Figure 4.3 visualizes the predicted output and actual output for testing data. From the figure, it can be realized that predicted output and actual output are practically identical.

TABLE 4.1: Performance comparison of McIT2FIS-EKF on non-linear system identification problem

	Algorithm	Rules	Train RMSE	Test RMSE	CPU Time (sec)
Type-1	eTS	19	-	0.008	-
	SAFIS	8	-	0.011	140
	PANFIS	27	0.011	0.0026	-
	GENEFIS	6	0.0004	0.0004	-
	McFIS	5	0.0023	0.0023	20
Type-2	SIT2FNN	3	-	0.0241	-
	McIT2FIS-GD	5	0.008	0.008	-
	McIT2FIS-EKF*	5	0.0009	0.0009	41.2

*PS = 85.09%

FIGURE 4.3: Actual and predicted output for non-linear system identification problem 1.

The effects of thresholds in *sample-learn-strategy* are demonstrated by examining the first 100 samples in the training phase. In Figure 4.4, the prediction error, adding threshold and update threshold are given. The spherical potential and novelty threshold are given in Figure 4.5. As observed from the figures, the samples whose prediction errors are higher than the adding threshold (above the red line in Fig. 4.4) and spherical potentials are lower than novelty threshold (below the red line in Fig. 4.5) are considered for rule growing. The samples falling between the adding threshold and the update threshold in Figure 4.4 are considered for rule updating, whereas, those below the update threshold are reserved for later learning stage. This mechanism enables the network to generalize the knowledge in the beginning and later to be fine-tuned, which has helped McIT2FIS-EKF approximate the underlying function accurately.

FIGURE 4.4: Demonstration of Learning Strategy with respect to Instantaneous Error, Adding Threshold and Update Threshold.

FIGURE 4.5: Demonstration of Learning Strategy with respect to Spherical Potential and Novelty Threshold.

4.5.2 Mackey-Glass Time Series Problem

Mackey-glass time series is one of standard problem in the literature, which has been described in Section 3.4.3. The aim of this problem is to predict $x(t + \hat{h})$

from n past values $[x(t), x(t - \Delta t), \dots, x(t - (n - 1)\Delta t)]$. For simulation study, Mackey-Glass parameters are set as follows $n = 4$, $\Delta t = 6$ and $\hat{h} = 85$. Three thousand samples are generated for the training process of the network. Five hundred samples are used for evaluating the performance of the predictive model. The predicted and actual output are plotted in Figure 4.6. It is clearly observed that the target has been accurately approximated. Table 4.2 describes the performance of McIT2FIS-EKF compared to different algorithms in the literature. For this problem, NDEI is employed as a criterion for doing benchmark. Compared to GENEFIS, the proposed algorithm achieves higher error reduction while employing one forth of the number of rules and 83.8% of training samples to approximate the system.

FIGURE 4.6: Actual and predicted output for Mackey-Glass time series problem-85.

TABLE 4.2: Performance comparison of McIT2FIS-EKF on Mackey-Glass time series problem-85

	Algorithm	Rules	Testing NDEI	CPU Time (sec)
Type-1	eTS	9	0.380	-
	SAFIS	6	0.376	-
	PANFIS	33	0.28	-
	GENEFIS	42	0.12	-
	McFIS	6	0.030	-
Type-2	SEIT2FNN	13	0.206	-
	McIT2FIS-GD	5	0.043	-
	McIT2FIS-EKF*	9	0.040	6.5

* $PS = 83.8\%$

4.6 Summary

A fully sequential learning algorithm for interval type-2 fuzzy inference system has been proposed in this chapter. The system based on an extended Kalman filtering method is capable of self-evolving its structure and parameters. In comparison to the iterative back-propagation learning algorithm of McIT2FIS-GD, the online parameter learning algorithm has shown its effectiveness in the generalization ability of the system. Comparative results have indicated the advantages of the proposed system over others state-of-the-art neuro-fuzzy inference systems in the literature.

In the next chapter, the system is extended with a modified memory structure to address renewable energy forecasting problems. The system is developed to quantify the uncertainty by constructing prediction intervals.

Chapter 5

Recurrent Neural Network Meta-cognitive Interval Type-2 Fuzzy Inference System

In the previous chapter, a sequential meta-cognitive interval type-2 fuzzy neural network has been proposed to handle temporally time-varying data. The algorithm employs a self-regulatory learning algorithm and an extended Kalman filtering based method to train the network. In this chapter, the learning algorithm of the proposed system is extended with a novel rule-based measure associated directly with prediction intervals for accurate forecasts of renewable energy. The IT2FIS architecture has been modified, and the algorithm is able to construct prediction intervals, which is indicated as recurrent neural network meta-cognitive interval type-2 fuzzy inference system, or RMcIT2FIS. Performance of RMcIT2FIS is evaluated using the real-world practical wind and wave data collected in Singapore.

5.1 Introduction

Penetration of renewable energy as an alternative source of power has increasingly captured the attention of researchers in forecasting. The nature of two fast-emerging renewable energy sources: the wind and wave, has shown that these sources are intermittent and uncertain. As a result, forecasting the sources at a certain state requires an evolving method that handles uncertainty. Moreover, as practical wind and wave are characterized by time series data, employing an infrastructure that memorizes a sequence of observations can provide better forecasting results and increase the performance of the system. The previous chapter has evaluated the sequential learning algorithm of an evolving interval type-2 neuro-fuzzy inference system. The meta-cognitive algorithm has clearly shown its generalization ability and fast training speed because it prevents redundant training samples for model updates and significantly reduces the number of training samples. Nonetheless, prediction interval (PI) is an uncharted territory in the current meta-cognitive learning and evolving fuzzy system (EFS) literature. While there exists a number of works on an offline estimation of prediction interval [13, 14, 39] which determines a lower and an upper bound and the confidence level that the target value lies within the bounds. PI is desirable in practice thanks to its capability of quantifying the uncertainty associated with the prediction and generating a qualitative forecasting information.

In this chapter, a meta-cognitive sequential learning algorithm for interval type-2 neuro-fuzzy inference system employing a novel recurrent neural network architecture in the output layer is proposed. The learning mechanism of the system is formulated on a neural fuzzy inference system. The network has five layers. The input layer passes data towards the membership layer to obtain Gaussian membership values. The firing layer calculates the strength of rules before employing interval reduction to approximate the output. The q factors [88] together with upper and lower weights enable the construction of two bounds in the output processing layer. In the output layer, two memory neurons are added to capture

the dynamics of time-series data. The algorithm with its ability to construct prediction intervals is indicated as recurrent neural network meta-cognitive interval type-2 fuzzy inference system, or RMcIT2FIS. Similarly, meta-cognitive learning mechanism as in other chapters controls the flow of RMcIT2FIS. To address the problem of prediction intervals, the algorithm in this chapter employs a criterion to measure the quality of PIs. The criterion is utilized by meta-cognition to facilitate the learning algorithm of the system in which the learning algorithm processes an input sample by deciding whether to delete it, learn it or reserve the sample based on its contained knowledge. These three decisions are *sample-delete-strategy*, *sample-learn-strategy* and *sample-reserve-strategy*, respectively.

The performance of proposed algorithm is measured based on real-world wave and wind prediction problems [21, 56]. The data of these renewable energy sources were collected in Singapore. For the wave data, prediction intervals are evaluated based on three wave characteristics, significant wave height, mean wave period and peak wave direction. The wind data is used in a wind speed prediction problem. The results are compared with other algorithms such as support vector regression, simplified interval type-2 fuzzy neural network (SIT2FNN)[18], projection-based learning meta-cognitive interval type-2 fuzzy inference (PBL-McIT2FIS)[92, 120] system and meta-cognitive interval type-2 fuzzy inference system gradient descent (McIT2FIS-GD)[56].

5.2 Recurrent Neural Network Meta-cognitive Interval Type-2 fuzzy Inference System

5.2.1 Problem Definition

Supposing that we are given the training data, $[(x(1), y(1)), \dots, (x(t), y(t)), \dots]$, where x is the input to the network and y is the corresponding target, without loss of generality, we assume that the input of the system consists of n features, $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ and the network has processed $t - 1$ samples and added

K rules. The objective of the algorithm is to estimate the underlying functional relationship $f[\cdot]$ so that the predicted output:

$$\hat{y}(t) = \frac{y_{lo}(t) + y_{up}(t)}{2} = f[\mathbf{x}(t), \boldsymbol{\theta}] \quad (5.1)$$

is as close as possible to the desired output, $y(t)$. It must be noted that $\boldsymbol{\theta}$ is the parameter vector of RMcIT2FIS, $y_{lo}(t)$ and $y_{up}(t)$ are the left and right output bound respectively. The error for t -th sample measuring the difference between the predicted output and the desired output (error $\mathbf{e}(t) = [e_l(t), e_r(t)]$) is given as:

$$\begin{aligned} e_l(t) &= y_{lo}(t) - y(t) \\ e_r(t) &= y_{up}(t) - y(t) \end{aligned} \quad (5.2)$$

The absolute prediction error is given by:

$$E(t) = \|\mathbf{e}(t)\| \quad (5.3)$$

5.2.2 Recurrent Neural Network Meta-cognitive Interval Type-2 fuzzy Inference System Architecture

The architecture of proposed recurrent neural network interval type-2 neuro-fuzzy inference system consists of five neural network layers. The system employs interval type-2 fuzzy sets in the rules and the consequent of each rule realizes Takagi-Sugeno-Kang fuzzy inference mechanism. The network structure is shown in Figure 5.1. The difference between this architecture and the one in previous chapters is that two memory neurons are developed in the output processing layer. This recurrent structure enables the network to learn the current knowledge and the knowledge at the previous step, which formulates the memorized feature of RMcIT2FIS. Assuming that the system approximate n input features and has grown K rules, the detailed description of the inference within the network at each layer, when the t - th sample arrives, is given as follow:

FIGURE 5.1: Architecture of the recurrent neural network interval type-2 fuzzy inference system.

Layer 1- Input layer: This layer contains n nodes as n input features. The inference of j – th node in this layer is given by:

$$u_j(t) = x_j(t); j = 1, 2, \dots, n \quad (5.4)$$

Layer 2- Membership layer: This layer calculates the upper and the lower membership strength of each j - th feature in every i - th rule. For clarification, the footprint of uncertainty is illustrated in Figure 5.2. The formulas are given by:

$$\mu_{ij}^{up}(t) = \begin{cases} \phi(m_1^{ij}, \sigma^i, u_j(t)) & u_j(t) < m_1^{ij} \\ 1 & m_1^{ij} \leq u_j(t) \leq m_2^{ij} \\ \phi(m_2^{ij}, \sigma^i, u_j(t)) & u_j(t) > m_2^{ij} \end{cases} \quad (5.5)$$

$$\mu_{ij}^{lo}(t) = \begin{cases} \phi(m_2^{ij}, \sigma^i, u_j(t)) & u_j(t) \leq \frac{m_1^{ij} + m_2^{ij}}{2} \\ \phi(m_1^{ij}, \sigma^i, u_j(t)) & u_j(t) > \frac{m_1^{ij} + m_2^{ij}}{2} \end{cases} \quad (5.6)$$

where,

$$\phi(m^{ij}, \sigma^i, u_j) = \exp\left(-\frac{(u_j - m^{ij})^2}{2(\sigma^i)^2}\right); i = 1, 2, \dots, K \quad (5.7)$$

where m_1, m_2 are the left and right center of the interval type-2 Gaussian membership function accordingly, and σ^i is the width of the i -th rule. The output of this layer can be represented by the intervals $[\mu_{ij}^{lo}(t), \mu_{ij}^{up}(t)]$.

FIGURE 5.2: Footprint of uncertainty using uncertain means and fixed standard deviation.

Layer 3- Firing layer: Each node in this layer represents the upper and lower firing strength of a rule. It is calculated by the algebraic product of membership values of a rule. The output of this layer is given as:

$$[f_i^{lo}(t), f_i^{up}(t)]; i = 1, 2, \dots, K \quad (5.8)$$

where,

$$f_i^{lo}(t) = \prod_{j=1}^n \mu_{ij}^{lo}(t) \quad (5.9)$$

and

$$f_i^{up}(t) = \prod_{j=1}^n \mu_{ij}^{up}(t) \quad (5.10)$$

Layer 4- Output processing layer: This layer employs memory neural network to process output bounds. Two recurrent neurons enable the algorithm to memorize the previous knowledge and generalize the current output. The outputs of the firing layer are combined with consequent weight parameters and the contribution of rules to provide the upper and lower bound of the output. The two bounds at time $t - th$ are $y_{lo}(t)$ and $y_{up}(t)$. The formulas for y_{lo} and y_{up} in this layer are given by:

$$y_{lo}(t) = \frac{(1 - q_l) \sum_{i=1}^K f_i^{up}(t) w_l^i + q_l \sum_{i=1}^K f_i^{lo}(t) w_l^i}{\sum_{i=1}^K (f_i^{lo}(t) + f_i^{up}(t))} + \beta_l y_{lo}^m(t) \quad (5.11)$$

$$y_{up}(t) = \frac{(1 - q_r) \sum_{i=1}^K f_i^{lo}(t) w_r^i + q_r \sum_{i=1}^K f_i^{up}(t) w_r^i}{\sum_{i=1}^K (f_i^{lo}(t) + f_i^{up}(t))} + \beta_r y_{up}^m(t) \quad (5.12)$$

where, w and q denote the consequent weight and control factor of the network as mentioned in the previous chapters. β_l and β_r are the respective weights of two memory signals y_{lo}^m and y_{up}^m . In this study, y_{lo}^m and y_{up}^m are initialized to zeros. These weight parameters controlling the influence of memory neurons on the final output are updated in the later learning stage of the algorithm. The processing

memory signals performed by memory neurons are given by:

$$y_{lo}^m(t) = \alpha_l y_{lo}(t-1) + (1 - \alpha_l) y_{lo}^m(t-1) \quad (5.13)$$

$$y_{up}^m(t) = \alpha_r y_{up}(t-1) + (1 - \alpha_r) y_{up}^m(t-1) \quad (5.14)$$

where, $\alpha_l, \alpha_r \in [0, 1]$ is the adaptive memory parameters of the memory neurons. α_l and α_r indicate the amount of knowledge in the last step for the current neurons to memorize and are also updated in the sequential learning algorithm.

Layer 4- Output processing layer: The crisp predicted output of the network is the combination of y_{lo} and y_{up} and is given as:

$$\hat{y}(t) = \frac{1}{2}(y_{lo}(t) + y_{up}(t)) \quad (5.15)$$

The above layers create a recurrent inference for RMcIT2FIS. The next section describes the sequential learning algorithm for the proposed network architecture. The objective is to approximate functional relationship between the input and output features based on fuzzy rules.

5.2.3 Self-regulatory Learning Algorithm for Recurrent Neural Network Meta-cognitive Interval Type-2 fuzzy Inference System

The learning algorithm employs meta-cognitive learning to optimize the parameters of the network. The task is performed by sequentially processing one by one sample in the training dataset. In this chapter, meta-cognitive learning mechanism employs three indicators to train the network. The prediction error $E(t)$ (equation (5.3)) is the first indicator that monitors the difference between the predicted output and the actual output. The second indicator is rule contribution measured by spherical potential, $\psi(t)$ [56, 67, 112]. It is calculated based on

the distance between the current sample and existing rules in the network in the hyper-dimensional space. $\psi(t)$ is the measure of novelty in the current sample. The formula of spherical potential employed in this chapter is given as:

$$\psi(t) = \sum_{i=1}^K \frac{f_i^{lo}(t) + f_i^{up}(t)}{2K} \quad (5.16)$$

where, K is the total number of contributing rules in the network. The third indicator is prediction interval (PI). This indicator monitors the knowledge by evaluating the relationship between the bounds and the desired output. It describes the certain confidence level the actual output falls into the predicted interval. PI is calculated by:

$$PI(t) = \frac{PI_{width}}{PI_{coverage}} \quad (5.17)$$

where, PI_{width} is the width of the interval and is given as $PI_{width} = |y_{lo}(t) - y_{up}(t)|$; $PI_{coverage}$ penalizes the intervals whose targets do not lie within the constructed bounds. $PI_{coverage}$ is calculated using Gaussian function with the mean $(y_{lo}(t) + y_{up}(t))/2$ and variance $(y_{lo}(t) - y_{up}(t))/2$. It is given by:

$$PI_{coverage} = \phi \left(y(t), \frac{y_{lo}(t) - y_{up}(t)}{2}, \frac{y_{lo}(t) + y_{up}(t)}{2} \right) \quad (5.18)$$

where, ϕ is shown in equation (5.7). PI is the direct measure of the quality of the constructed interval, i.e, a higher PI indicates that the distance between the upper and lower bound is wide and the desired output falls further to the side than into the interval. Hence, the constructed interval is low-quality, which allows the network to adapt its parameters. The format of the indicator PI is similar to the prediction error in term of measuring knowledge in data. In the next section, meta-cognitive learning mechanism monitoring the above mentioned indicators

and controlling the structure adaption is presented, followed by the performance study in practical wave and wind prediction problem.

5.2.3.1 Sample Delete Strategy

The sample is deleted if its knowledge is redundant to the network. In other words, its prediction error and prediction interval are lower than the delete threshold, E_d and PI_d . E_d and PI_d are set close to zero. According to the equation (5.17), the lower value of PI results in a higher coverage and better fitting width of the constructed interval. In all the problems studied in this chapter, E_d is set in the range [0.001, 0.01] and PI_d is set in the range [0.01, 0.1]. The conditions for *sample-delete-strategy* are given as:

$$E(t) < E_d \quad \text{AND} \quad PI(t) < PI_d \quad (5.19)$$

5.2.3.2 Sample Learn Strategy

This strategy works by either growing a new rule or adapting the parameters of the contributing rules in the system. Meta-cognition monitors the knowledge contained in the current sample and makes the decision on how-to-learn the sample. The adding decision is described in the following section.

1. The Rule Adding Criterion:

If the sample contains significant knowledge, a new rule is grown in the network. In other words, if the prediction error and the prediction interval of the sample are higher than the adding thresholds, E_a and PI_a respectively, a new rule is added to the network. In addition, the novelty in the current sample is also assessed. The contained knowledge is novel if the spherical potential is lower than the novelty threshold, E_S . The conditions are as follows:

$$E(t) > E_a \quad \text{AND} \quad PI(t) > PI_a \quad \text{AND} \quad \psi(t) < E_S \quad (5.20)$$

For the problems considered in this chapter, E_a is set in the range [0.15, 0.5], PI_a is set in the range [0.2, 0.8] and E_S is set in the range [0.01, 0.6]. After growing the new rule, the three thresholds are self-adaptive as:

$$E_a := (1 - \gamma_1)E_a + \gamma_1 E(t) \quad (5.21)$$

$$PI_a := (1 - \gamma_1)PI_a + \gamma_1 PI(t) \quad (5.22)$$

$$E_S := (1 - \gamma_2)E_S - \gamma_2 \psi(t) \quad (5.23)$$

where, γ_1 and γ_2 are the control slope parameters that E_a and PI_a increase and E_S decrease. The aim of self-adaptive control slope parameters in this algorithm is to let the system capture the general knowledge at the beginning stage of the learning process and later fine-tune it. The two parameters γ_1 and γ_2 are set close to zero. The new $(K + 1) - th$ rule initialization of the centers, the width, and the new weights are adopted from the equation (4.19) to (4.21).

2. The Rule Updating Criterion:

If the prediction error and prediction interval are not high enough to grow a rule, meta-cognition considers the sample for updating the rules in the network. The criterion is as follows:

$$\begin{aligned} E_u < E(t) < E_a \quad \text{AND} \quad PI_u < PI(t) < PI_a \\ \text{AND} \quad \psi(t) > E_S \end{aligned} \quad (5.24)$$

where, E_u and PI_u are the updating threshold of prediction error and prediction interval respectively. E_u is set in the range [0.01, 0.04] and PI_u is

set in the range $[0.15, 0.5]$ in all the problems studied in this chapter. After updating the network, the three thresholds are self-adaptive as:

$$E_u := (1 - \gamma_1)E_u - \gamma_1 E(t) \quad (5.25)$$

$$PI_u := (1 - \gamma_1)PI_u - \gamma_1 PI(t) \quad (5.26)$$

$$E_S := (1 - \gamma_2)E_S + \gamma_2 \psi(t) \quad (5.27)$$

In contrast to the rule growing strategy, E_u and PI_u are designed to decrease and the novelty threshold E_S is to increase. The self-adaptive features of these thresholds trigger the learning of samples with significant knowledge at the beginning and later fine-tune the system for better generalization. The parameter updating algorithm of RMcIT2FIS is based on the extended Kalman filtering method described in detail in Chapter 4. The parameter vector of the network is $\theta = [\mathbf{m}_1, \mathbf{m}_2, \sigma, \mathbf{w}_l, \mathbf{w}_r, \mathbf{q}, \alpha, \beta]$. θ is the z -dimensional vector and $z = K(2n + 3) + 6$. The updating formulas of θ are derived from the equation (4.25) to (4.29). For the gradient of predicted output with respect to α and β , the derivations are as follow:

$$\frac{\partial \hat{y}(t)}{\partial \beta_l} = \frac{1}{2} y_{lo}^m(t) \quad (5.28)$$

$$\frac{\partial \hat{y}(t)}{\partial \beta_r} = \frac{1}{2} y_{up}^m(t) \quad (5.29)$$

$$\frac{\partial \hat{y}(t)}{\partial \alpha_l} = \frac{1}{2} \beta_l (y_{lo}(t-1) - y_{lo}^m(t-1)) \quad (5.30)$$

$$\frac{\partial \hat{y}(t)}{\partial \alpha_r} = \frac{1}{2} \beta_r (y_{up}(t-1) - y_{up}^m(t-1)) \quad (5.31)$$

5.2.3.3 Sample Reserve Strategy

The sample is reserved for training in the later stage if its contained knowledge is not recognized by the above-mentioned strategies. As mentioned earlier in the sample add and update section, the samples are only learnt if the measured learning criteria are satisfied. Otherwise, the sample is reserved for learning at a later stage.

The three learning strategies, *sample-delete-strategy*, *sample-learn-strategy* and *sample-reserve-strategy* formulate self-regulatory meta-cognitive learning algorithm of the proposed IT2FIS. They are respectively three decision on *what-to-learn*, *how-to-learn* and *when-to-learn* in the sequential learning algorithm. In the next section, we perform the evaluation of propose algorithm on practical wave and wind prediction problems.

5.3 Performance on Wave Energy Characteristics

In this section, we shall discuss the effectiveness of the proposed algorithm by constructing prediction intervals. The performance of RMcIT2FIS is evaluated on the waverider data collected from the buoys located offshore Semakau Island, Singapore. The waverider data is discussed in the following subsection.

5.3.1 Waverider Data

For experimental study, the waverider data are collected from an instrument in the area offshore Semakau Island, Singapore. Directional wave buoys (Figure 5.3) were deployed to record wave statistics for 30 minutes intervals including the number of zero-crossings, wave height, wave period and wave direction. The waveriders of this project are the GPS-based directional buoys mentioned in [121, 122].

FIGURE 5.3: Directional waverider buoy with integrated high capacity cell and navigation light.

A single GPS receiver is used to serve for buoy positioning and facilitate waves measurement as well as buoy retrieval. In order to store the collected data, the standard integrated data logger is used, which is empowered by a high capacity primary cell. The cell operates under all wave conditions and weather circumstances for a long period of time without replacement. The waverider measuring unit is incorporated with motion sensors, which gives out the real-time heave, the direction as well as the common wave parameters like wave height and wave period. The measuring unit is built around stabilized platform sensor so that wave height measurement is enabled by a single accelerometer. The wave direction is computed by measuring direct pitch and roll in combination with horizontal accelerometers and a compass. Any magnetic parts that may interfere the magnetic compass and wave direction measurement need to be kept away in the installation of measuring unit. The unit may not be exposed to seawater and must be installed in a watertight compartment. Additionally, the waverider features high frequency (HF) transmitter with range of 50 kilometers over sea and a directional receiving antenna integrated LED flash for better visibility to sea vehicles.

Wave measurement data consists of four features: significant wave height($Hm0$), peak wave period(Tp), zero-upcrossing wave period(Tz) and peak wave direction($Dirp$). Significant wave height is the average height of the highest one-third of the waves in a wave spectrum. Peak wave period is the wave period with the most significant wave component, while the mean wave period is the mean of all the components contributing to wave period. Mean wave direction is defined as the mean of all the individual wave directions. The studied time period of this wave measurement project is from October 2014 to September 2015. Figure 5.4 shows four wave characteristics during the first month of the experiment. The pattern of daily variation in the significant wave height can be easily observed, whereas the other characteristics fluctuate stronger, especially peak wave period. A number of extreme values can be seen, i.e., the peak wave period suddenly goes up to forty seconds and drops quickly. In the wave direction and zero-upcrossing period data of the first month, the differences between peak and low values are

FIGURE 5.4: Wave characteristics (Oct 2014 - Nov 2014).

not substantial even though there are still some periods when the direction irregularly experiences immediate rises and falls . The below subsection discusses the performance of RMcIT2FIS on the above-mentioned wave data.

5.3.2 Performance Measures

In order to evaluate the performance of the prediction interval, coverage probability is employed to indicate the percentage of samples covered by the upper and lower bounds of the intervals. Prediction intervals coverage probability (PICP), as used in [13, 14, 29] is defined as:

$$PICP = \frac{1}{N} \sum_i^N \in_i \quad (5.32)$$

where N is the number of samples and $\in_i$ show the coverage of PIs towards the $i - th$ sample. $\in_i$ is defined as:

$$\in_i = \begin{cases} 1, & \text{if } y_i \in [L_i, U_i] \\ 0, & \text{if } y_i \notin [L_i, U_i] \end{cases} \quad (5.33)$$

where L_i and U_i are lower bound and upper bound, respectively.

In addition to PICP, the quality of prediction intervals is analyzed on another criterion, the width of prediction intervals. In the literature, a measure of the width is defined as the normalized average value of PIs, or PINAW [123]. The prediction intervals are considered high-quality if they are able to cover most of the samples (or give a high PICP) while providing low widths. In this thesis, we study the performance of prediction intervals by employing the width defined as δ together with y_{lo} and y_{up} . The prediction intervals then shall be examined by monitoring PICP. The lower and upper are given as:

$$y_{lo} = y - \delta, y_{up} = y + \delta \quad (5.34)$$

5.3.3 Significant Wave Height

The number of samples for training data is 1200, which was collected every half an hour in the first month. For comparison purpose, data is normalized in the range $[-1, 1]$. The main objective is to predict the wave height half-an-hour ahead $y(t+1)$ given the past five data samples $[y(t), y(t-1), y(t-2), y(t-3), y(t-4)]$. Figure 5.5 shows the constructed prediction intervals for testing set regarding 10% of PIs width. It can be noticed that the PIs cover test samples by a high percentage. The test data normally lies between upper bound and lower bound in most of the cases. The peak wave height, low wave height and fluctuated periods are also well generalized although there is one period that is not satisfactorily covered by the constructed PIs. The width of PIs is determined based on the uncertainty of the data set. Lower uncertainty in the data means that the width of PIs could be set narrower and vice versa. In order to study the plausibility of the algorithm, different values of PIs width are used in the experiment. The coverage probability, the width of PIs and construction times of training and testing data set are shown in Table 5.1.

The results in Table 5.1 imply the adaptive ability of the system under a certain value of PI width. An adequate PICP is certainly achieved with the reasonable range of the width. In this case of significant wave height, PIs with coverage probability approximately 90% are acquired when the width is higher than 20% of confidence. In addition, the computationally fast PIs construction indicates the plausibility of RMcIT2FIS.

5.3.4 Mean Wave Period

Together with the wave height, period plays an important role in representing a certain sea state. Mean wave period as a critical feature controls the average of every single wave period in a time-series. In this section, we conduct an experiment to study PIs with 30-minute forecasting horizon with respect to mean period.

FIGURE 5.5: Prediction intervals for significant wave height corresponding to 10% width.

TABLE 5.1: Performance of PIs for significant wave height

PI Width (δ)	Training PICP	Testing PICP	Times (s)
0.05	0.63	0.43	0.8
0.10	0.88	0.59	0.8
0.15	0.96	0.79	0.7
0.20	0.99	0.87	0.9
0.25	0.99	0.93	0.8

Features and data selection are identical to the study of significant wave height, i.e., five historical periods are utilized to predict the half-an-hour ahead wave period. Figure 5.6 shows the upper and lower bounds and actual wave period for

testing dataset corresponding to 10% PIs width. It can be observed from the figure that the fluctuation in mean wave period data is stronger than in significant wave height. As a result, the width of PIs becomes wider to achieve a respective coverage. To obtain a comparative coverage, the width needs to expand approximately 5% more than that of the wave height. However, when the expansion reaches relatively 25%, the constructed prediction intervals are likely to provide adequate coverage over the testing dataset.

FIGURE 5.6: Prediction intervals for mean wave period corresponding to 10% width.

The constructed PIs for mean wave period under different preassigned values of δ are described in Table 5.2. It can be seen from the table that a 90% prediction intervals could be achieved if the width of PIs is higher than 20%. This level of the width is understandable due to the fact that zero-upcrossing period is more

irregular than the wave height. To better understand the performance of the algorithm on wave energy characteristics, we conduct experiments on the peak wave direction in the next section.

TABLE 5.2: Performance of PIs for mean wave period

PI Width (δ)	Training PICP	Testing PICP	Times (s)
0.10	0.79	0.43	0.7
0.15	0.92	0.63	0.8
0.20	0.97	0.86	0.8
0.25	0.99	0.92	0.8

5.3.5 Peak Wave Direction

In this subsection, we study another characteristic of wave measurement data, namely peak wave direction. Similar to the previous experiments, one-step-ahead direction is predicted using historical data. The direction of a certain wave state can be extremely useful in forecasting. It is also known to be the most uncertain feature because the wave can suddenly fluctuate completely in an opposite direction. Hence, in order to quantify the wave direction, it is reasonable to choose a higher PIs width. In this study, the width of PIs is preassigned as 30% and 35%. The performance is then evaluated based on the coverage probability. Table 5.3 shows the prediction intervals evaluation indices for training and testing samples. As can be seen from the table, in order to obtain a high coverage probability of practically 93% in the test case, the width of PIs is at 35%. This result strongly shows the uncertainty of wave direction. Additionally, the PI construction time in this experiment is notably longer than the previous experiments since the algorithm employs more rules to estimate the underlying functional relationship between the input and output.

TABLE 5.3: Performance of PIs for peak wave direction

PI Width (δ)	Training PICP	Testing PICP	Times (s)
0.30	0.87	0.85	3.5
0.35	0.92	0.93	3.4

5.4 Performance on Wind Speed Prediction Problem

In this section, the proposed algorithm is evaluated based on the wind speed prediction problem. The performance of RMcIT2FIS is compared with other algorithms in the literature including Support vector regression (SVR), SIT2FNN [18], PBL-McIT2FIS [92] and McIT2FIS-GD [56]. To study the effect of prediction interval, RMcIT2FIS and McIT2FIS-EKF are employed on a practical forecasting problem, namely wind speed prediction. The performances of the algorithms are evaluated in Matlab R2013b environment on a Window system with Xeon CPU and 16GB RAM.

5.4.1 RMcIT2FIS for Wind Speed Point Forecast

RMcIT2FIS is implemented for the wind speed prediction task in this study with wind data adopted from Section 3.4.5. As mentioned earlier, the training data consist of five hundred samples, in each of which ten features are extracted. The aim is to predict the future wind speed based on historical values. The root mean square error and number of rules for this problem are described in Table 5.4. It is observed that RMcIT2FIS is able to forecast the output with low error and a reasonable number of rules. The percentage of samples used for training is 73%.

5.4.2 RMcIT2FIS for Wind Speed Prediction Intervals

In this section, we construct prediction intervals for the above-mentioned wind speed prediction problem. In order to evaluate the performance of the prediction

Algorithm	Rules	Training RMSE	Testing RMSE
SVR	-	0.197	0.214
SIT2FNN	4	0.194	0.187
PBL-McIT2FIS	4	0.110	0.161
McIT2FIS-GD	4	0.136	0.132
McIT2FIS-EFK	4	0.135	0.141
RMcIT2FIS*	4	0.1336	0.1381

*PS = 73%

TABLE 5.4: Performance comparison on wind speed prediction problem

interval, coverage probability and average width are employed. Prediction intervals coverage probability (PICP) has been described in the previous section and the average width of the intervals, PIAW [123] is given as:

$$PIAW = \frac{1}{N} \sum_{i=1}^N \frac{(U_i - L_i)}{R} \quad (5.35)$$

where L_i and U_i are lower bound and upper bound, respectively; R is the range of the underlying targets (maximum minus minimum). Figure 5.7 shows the constructed prediction intervals for testing data set. It is observed that the majority of actual test data falls into the lower upper bound. For testing data set, the studied prediction intervals cover 92% of samples while the average width is 34.5%. In this problem, the underlying trend has been well approximated by the upper and lower bound, and the shapes of the two lines can generalize the data. However, there exist the peak periods which are not properly covered by the constructed intervals. It is also noticed that the width of PIs tends to expand in the valley and scale down in the peak. Table 5.5 shows PICP, PIAW for training, testing data the percentage of samples employed. From the table, we can notice that RMcIT2FIS is able to give an effective quantification for wind speed prediction problem while

learning 67.8% of samples in a sequential mechanism.

FIGURE 5.7: Prediction intervals for wind speed problem.

TABLE 5.5: PI evaluation for training and testing

	Training	Testing
PICP	0.81	0.92
PIAW	0.24	0.35

* $PS = 67.8\%$

5.5 Summary

A recurrent neural network meta-cognitive interval type-2 fuzzy inference system using associative memory neurons to cope with time-series data has been developed in this chapter. The learning algorithm addressing prediction intervals as a

measure to quantify uncertainty in data is capable of self-regulating its learning process based on the contained knowledge of the data sample. The proposed system has been evaluated on wave energy characteristics prediction and wind speed prediction problem. Experimental results have indicated advantages of the proposed system in handling and quantifying uncertainty. In the next chapter, this thesis is summarized and potential research directions will be presented.

Chapter 6

Summary and Future Works

In the previous chapters, the learning algorithm for interval type-2 neuro-fuzzy inference system has been extended using different parameters updating and structure growing mechanisms. This chapter summarizes the key contributions of this thesis followed by potential future directions.

6.1 Thesis Summary

This thesis develops an evolving meta-cognitive interval type-2 neuro-fuzzy inference system to handle and quantify uncertainty in time-varying data by addressing prediction intervals. The main contributions of the thesis are concluded in the following key points:

- A meta-cognitive interval type-2 neuro-fuzzy inference system and its gradient descent learning algorithm. The proposed system is able to handle non-stationarity in data effectively by employing interval type-2 fuzzy sets.
- An evolving sequential learning algorithm to handle time-varying data that arrives in sequence. The algorithm selects appropriate samples for training and employs a fast learning algorithm to adapt the network structure.
- A recurrent neural network structure for interval type-2 neuro-fuzzy inference system to address prediction intervals. The system is developed with memory neurons and the algorithm monitors a PI-specific criterion for learning.
- Construction of prediction intervals for renewable energy. The proposed system is capable of providing prediction intervals for wave measurement and wind energy data.

The contributions are summarized in the following subsections.

6.1.1 Meta-cognitive Interval Type-2 Fuzzy Inference System and Its Gradient Descent Learning Algorithm

An interval type-2 fuzzy inference system and its gradient descent learning algorithm have been proposed (McIT2FIS-GD). The system is able to handle uncertainty and evolve to cope with time-varying data. The inference employs a fast

interval type-reduction and the learning algorithm is based on an iterative gradient descent algorithm. The proposed McIT2FIS-GD is capable of approximating the underlying functional relationship in data and its principles of meta-cognition have enabled the system to avoid over-fitting and facilitated the generalization of the system.

The performance of McIT2FIS-GD has been evaluated on a set of function approximation problems and a practical wind speed prediction problem. The experimental results on a set of function approximation problems compared to other state-of-the-art neuro-fuzzy systems in the literature have shown that the system achieves significantly lower prediction error while growing a lesser number of rules. In case of the wind energy data set, comparative results have indicated that McIT2FIS-GD is capable of forecasting the functional relationship between the past and the future wind speeds. The use of meta-cognition has helped McIT2FIS-GD obtain better accuracy.

6.1.2 An Evolving Sequential Learning Algorithm for Interval Type-2 Neuro-Fuzzy Inference System

For the second contribution, an evolving sequential learning algorithm for interval type-2 neuro-fuzzy inference system has been proposed to handle time-varying data that arrives in sequence (McIT2FIS-EKF). The system processes each arriving data once and discards it afterward. The data is learnt by a self-regulatory learning mechanism and an extended-Kalman filtering method.

The performance is evaluated on a benchmark system identification problem and a time-series prediction problem. For the system identification problem, the one-step-ahead output has been estimated accurately. The system obtains an error of one order less than the compared algorithm, including SIT2FIS [18], GENEFIS [95], SAFIS [73] and McFIS [67]. In case of the time series problem, the system provides an error reduction compared to other type-2 neuro-fuzzy systems. The

remarkable results indicate that the proposed sequential learning algorithm can generalize the underlying trend in data with accuracy.

6.1.3 Recurrent Structure for Interval Type-2 Neuro-Fuzzy Inference System

The sequential learning algorithm has been extended by employing a memory structure developed with recurrent neural network to cope with time-series data. The system is capable of memorizing the previous knowledge to better generalize the future target. The learning algorithm measures a prediction interval criterion to control the rule growing, rule updating in the network. The proposed algorithm is capable of solving function approximation problems and providing output as either crisp targets or prediction intervals.

6.1.4 Prediction Intervals for Renewable Energy Time-Series Data

In order to quantify the forecasts, prediction intervals have been constructed and evaluated based on the real-world wave and wind prediction problems. For the wave prediction problem, wave measurement data was collected from the buoys deployed in our project located offshore Semakau Island, Singapore during the period of one year from October 2014 to September 2015. Prediction intervals for significant wave height, mean wave period and peak wave direction have been constructed for verification. The results have indicated that constructed prediction intervals have a good coverage probability of approximately 90% of total testing samples with 20% to 30% confidence. In case of wind speed prediction problem, the algorithm is able to estimate the underlying functional relationship in data and construct sufficient prediction intervals. The performance evaluation in the two practical problems has indicated that the uncertainty has been successfully quantified using prediction intervals and the forecasting results can be considered to improve the decision-making process.

6.2 Future Directions

The research in this thesis could be extended in various potential directions. They are described in the following subsections.

6.2.1 Improvements in Learning Algorithm

- **Objective Functions:**

The proposed algorithms in this thesis employ prediction error-based cost function. However, in a prediction intervals problem, the prediction intervals-based objective functions directly address the associated issues. In the literature, PIs-based cost functions as coverage width-based functions have been proposed in [13, 14, 39]. Unlike prediction error-based cost function, the coverage width-based functions constrain the width and the coverage of the intervals to optimize the error. These PIs-based cost function should be studied in the future for better performance.

- **Verification of Rules:**

The meta-cognitive learning mechanism is effective in the generalization of the system and the rules are grown as the training data arrives. The rules centers and widths are initialized with the initial means and variances respectively and they are adaptive over time. Due to that automatic mechanism, the rules may contain incorrectness. The reasons possibly come from the training data itself or in the noise and vagueness in data. Therefore, one future direction of this study is to develop a method to verify the rule at the end of the training phase. This could be done and advised by expert experience as in [124] or by employing rules pruning mechanisms proposed in [67]. The verification of rules should be considered to improve the performance of the algorithm.

- Classification problems:

The proposed algorithm has been verified using a set of benchmark regressions and time-series. It should be extended to address classification problems. In order to solve these problems, class-specific criterion needs to be considered for both the inference and the learning mechanism of the system [56, 67]. As part of future work, the self-regulate learning algorithm can be applied for various benchmark classification problems. Its characteristics can be developed for better performance.

6.2.2 Improvements in Renewable Energy Forecasting

- Long-Term Forecasting:

It has been shown in the literature that short-term forecasting is important for day-to-day operation of the electrical power systems [4]. It shows impacts on online-scheduling and security function of an energy management system. On the other hand, longer-term forecasting cares for the unusual situations in the whole system such as peak and drop electricity demand for capacity planning, which is crucial for the management and maintenance of the system [5]. As a result, one future direction of this research is to develop the method with wider forecasting horizon or longer time in advance. From that point of view, the scheduling functions are constructed more precisely, being aware of all the short-term and long-term components. The long-term forecast could be produced by employing parallel and series model discussed in [125], where either the actual or predicted output is fed back into the model to provide the long-step ahead target.

- Data Selection and Processing:

As renewable energy is growing significantly in the future, the data is consequently increasing dramatically, which required critical data selection and feature extraction for an effective forecasting. Instead of directly using historical features, one can employ multiple types of data combining different

renewable energy sources. These renewable energy sources having strong correlations due to their physical originated nature need to be thoroughly studied. With the consideration on data, the performance of renewable energy forecasting methods would be certainly improved.

Author's Publication

Journal

1. A. K. Das, Nguyen Anh, Sundaram Suresh, and N. Srikanth. An interval type-2 fuzzy inference system and its meta-cognitive learning algorithm. *Evolving Systems*, 7(2):95105, 2016. ISSN 1868-6486.
2. Nguyen Anh, Sundaram Suresh, M. Pratama and N. Srikanth. Interval Prediction of Wave Energy Characteristics using Meta-cognitive Interval Type-2 Fuzzy Inference System. *Knowledge-Based System* (Revision Submitted).

Conference

1. Nguyen Anh, Ye Ren, Sundaram Suresh, and N. Srikanth. Meta-cognitive Interval Type-2 Fuzzy Neural Network for Significant Wave Height Forecasting. *Asian Wave and Tidal Energy Conference (AWTEC)*, 2016.
2. Nguyen Anh, Mukesh Prasad, Sundaram Suresh. Wind Speed Intervals Prediction using Meta-cognitive Approach. *International Neural Network Society Conference on Big Data and Deep Learning (INNS-BDDL)*, 2018.
3. Nguyen Anh, Mukesh Prasad, Sundaram Suresh. Wave Forecasting using Meta-cognitive Interval Type-2 Fuzzy Inference System. *International Neural Network Society Conference on Big Data and Deep Learning (INNS-BDDL)*, 2018.

Bibliography

- [1] E. Lannoye. Renewable energy integration: practical management of variability, uncertainty, and flexibility in power grids [book reviews]. *IEEE Power and Energy Magazine*, 13(6):106–107, Nov 2015. ISSN 1540-7977.
- [2] P. E. McSharry, S. Bouwman, and G. Bloemhof. Probabilistic forecasts of the magnitude and timing of peak electricity demand. *IEEE Transactions on Power Systems*, 20(2):1166–1172, May 2005. ISSN 0885-8950.
- [3] E. Gonzalez-Romera, M. A. Jaramillo-Moran, and D. Carmona-Fernandez. Monthly electric energy demand forecasting based on trend extraction. *IEEE Transactions on Power Systems*, 21(4):1946–1953, Nov 2006. ISSN 0885-8950.
- [4] J. W. Taylor and P. E. McSharry. Short-term load forecasting methods: An evaluation based on european data. *IEEE Transactions on Power Systems*, 22(4):2213–2219, Nov 2007. ISSN 0885-8950.
- [5] D. Srinivasan and M. A. Lee. Survey of hybrid fuzzy neural approaches to electric load forecasting. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, volume 5, pages 4004–4008 vol.5, Oct 1995.
- [6] Juan M. Morales, Antonio J. Conejo, Henrik Madsen, Pierre Pinson, and Marco Zugno. *Renewable Energy Sources—Modeling and Forecasting*, pages 15–56. Springer US, Boston, MA, 2014. ISBN 978-1-4614-9411-9.

- [7] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos. Long-term wind speed and power forecasting using local recurrent neural network models. *IEEE Transactions on Energy Conversion*, 21(1):273–284, March 2006. ISSN 0885-8969.
- [8] R. Donida Labati, A. Genovese, V. Piuri, F. Scotti, and G. Sforza. A decision support system for wind power production. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP(99):1–15, 2018. ISSN 2168-2216.
- [9] Y. Wang, Y. Shen, S. Mao, G. Cao, and R. M. Nelms. Adaptive learning hybrid model for solar intensity forecasting. *IEEE Transactions on Industrial Informatics*, PP(99):1–1, 2018. ISSN 1551-3203.
- [10] F. Golestaneh, P. Pinson, and H. B. Gooi. Very short-term nonparametric probabilistic forecasting of renewable energy generation x2014; with application to solar energy. *IEEE Transactions on Power Systems*, 31(5):3850–3863, Sept 2016. ISSN 0885-8950.
- [11] Hoang-Jyh Leu, Chao-Chan Wu, and Chiu-Yue Lin. Technology exploration and forecasting of biofuels and biohydrogen energy from patent analysis. *International Journal of Hydrogen Energy*, 37(20):15719 – 15725, 2012. ISSN 0360-3199. The 2011 Asian Bio-Hydrogen and Biorefinery Symposium (2011ABBS).
- [12] Gordon Reikard. 7 - wave energy forecasting. In George Kariniotakis, editor, *Renewable Energy Forecasting*, Woodhead Publishing Series in Energy, pages 199 – 217. Woodhead Publishing, 2017. ISBN 978-0-08-100504-0.
- [13] Hao Quan, D. Srinivasan, and A. Khosravi. Construction of neural network-based prediction intervals using particle swarm optimization. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2012.

- [14] H. Quan, D. Srinivasan, and A. Khosravi. Short-term load and wind power forecasting using neural network-based prediction intervals. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):303–315, 2014.
- [15] C. F. Juang and C. Y. Chen. Data-driven interval type-2 neural fuzzy system with high learning accuracy and improved model interpretability. *IEEE Transactions on Cybernetics*, 43(6):1781–1795, Dec 2013. ISSN 2168-2267.
- [16] Chi-Hsu Wang, Chun-Sheng Cheng, and Tsu-Tian Lee. Dynamical optimal training for interval type-2 fuzzy neural network (t2fnn). *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(3):1462–1477, 2004.
- [17] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak. Extended kalman filter based learning algorithm for type-2 fuzzy logic systems and its experimental evaluation. *IEEE Transactions on Industrial Electronics*, 59(11):4443–4455, Nov 2012. ISSN 0278-0046.
- [18] Y. Y. Lin, S. H. Liao, J. Y. Chang, and C. T. Lin. Simplified interval type-2 fuzzy neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):959–969, May 2014. ISSN 2162-237X.
- [19] S.W. Tung, C. Quek, and C. Guan. et2fis: An evolving type-2 neural fuzzy inference system. *Information Sciences*, 220(Supplement C):124 – 148, 2013. ISSN 0020-0255. Online Fuzzy Machine Learning and Data Mining.
- [20] C. F. Juang and Y. W. Tsao. 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.
- [21] A. K. Das, S. Suresh, and N. Srikanth. Meta-cognitive interval type-2 neuro-fuzzy inference system for wind prediction. In *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pages 1–6, 2014.

- [22] M. Lydia and S. Suresh Kumar. A comprehensive overview on wind power forecasting. In *2010 Conference Proceedings IPEC*, pages 268–273, Oct 2010.
- [23] Chris Chatfield. *Time-Series Forecasting*. Chapman and Hall/CRC, October 25, 2000. ISBN 9781584880639.
- [24] D. C. Hill, D. McMillan, K. R. W. Bell, and D. Infield. Application of autoregressive models to u.k. wind speed data for power system impact studies. *IEEE Transactions on Sustainable Energy*, 3(1):134–141, Jan 2012. ISSN 1949-3029.
- [25] S.S. Haykin. *Neural Networks: A Comprehensive Foundation*. International edition. Prentice Hall, 1999. ISBN 9780139083853.
- [26] Laurene V. Fausett. *Fundamental of Neural Networks: Architectures, Algorithms, and Applications*. 12 1993. ISBN 0133341860.
- [27] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965. ISSN 0019-9958.
- [28] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning-iii. *Information Sciences*, 9(1):43 – 80, 1975. ISSN 0020-0255.
- [29] A. Khosravi, S. Nahavandi, and D. Creighton. Prediction intervals for short-term wind farm power generation forecasts. *IEEE Transactions on Sustainable Energy*, 4(3):602–610, July 2013. ISSN 1949-3029.
- [30] C Monteiro, Ricardo Bessa, V Miranda, Audun Botterud, Jianhui Wang, G Conzelmann, , Decision Information Sciences, and INESC Porto. Wind power forecasting: State-of-the-art 2009. 11 2009.

- [31] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Transactions on Neural Networks*, 22(3):337–346, March 2011. ISSN 1045-9227.
- [32] J. T. Gene Hwang and A. Adam Ding. Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, 92(438): 748–757, 1997. ISSN 01621459.
- [33] Richard D. de Veaux, Jennifer Schumi, Jason Schweinsberg, and Lyle H. Ungar. Prediction intervals for neural networks via nonlinear regression. *Technometrics*, 40(4):273–282, 1998. ISSN 00401706.
- [34] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995. ISBN 0198538642.
- [35] B. Efron. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7(1):1–26, 01 1979.
- [36] D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, volume 1, pages 55–60 vol.1, 1994.
- [37] A. Khosravi, S. Nahavandi, and D. Creighton. Construction of optimal prediction intervals for load forecasting problems. *IEEE Transactions on Power Systems*, 25(3):1496–1503, 2010.
- [38] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. W. C. van Lint. Prediction intervals to account for uncertainties in travel time prediction. *IEEE Transactions on Intelligent Transportation Systems*, 12(2): 537–547, June 2011. ISSN 1524-9050.
- [39] H. Quan, D. Srinivasan, A. Khosravi, S. Nahavandi, and D. Creighton. Construction of neural network-based prediction intervals for short-term

- electrical load forecasting. In *2013 IEEE Computational Intelligence Applications in Smart Grid (CIASG)*, pages 66–72, April 2013.
- [40] J. Varanasi and M. M. Tripathi. Artificial neural network based wind power forecasting in belgium. In *2016 IEEE 7th Power India International Conference (PIICON)*, pages 1–6, Nov 2016.
- [41] Farivar Fazelpour, Negar Tarashkar, and Marc A. Rosen. Short-term wind speed forecasting using artificial neural networks for tehran, iran. *International Journal of Energy and Environmental Engineering*, 7(4):377–390, Dec 2016. ISSN 2251-6832.
- [42] N. Amjady, F. Keynia, and H. Zareipour. Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization. *IEEE Transactions on Sustainable Energy*, 2(3):265–276, July 2011. ISSN 1949-3029.
- [43] M. Abuellla and B. Chowdhury. Solar power forecasting using artificial neural networks. In *2015 North American Power Symposium (NAPS)*, pages 1–5, Oct 2015.
- [44] Deepthi.I. Gopinath and G.S. Dwarakish. Wave prediction using neural networks at new mangalore port along west coast of india. *Aquatic Procedia*, 4:143 – 150, 2015. ISSN 2214-241X. INTERNATIONAL CONFERENCE ON WATER RESOURCES, COASTAL AND OCEAN ENGINEERING (ICWRCOE’15).
- [45] S. N. Londhe and Vijay Panchang. One-day wave forecasts based on artificial neural networks. *Journal of Atmospheric and Oceanic Technology*, 23(11):1593–1603, 2006.
- [46] Deepthi I Gopinath and GS Dwarakish. Real-time prediction of waves using neural networks trained by particle swarm optimization. *The International Journal of Ocean and Climate Systems*, 7(2):70–79, 2016.

- [47] Hideyuki Takagi. Fusion technology of neural networks and fuzzy systems: A chronicled progression from the laboratory to our daily lives. 10, 01 2000.
- [48] Isao HAYASHI and Motohide UMANO. Perspectives and trends of fuzzy-neural networks. *Journal of Japan Society for Fuzzy Theory and Systems*, 5(2):178–190, 1993.
- [49] N. N. Karnik, J. M. Mendel, and Qilian Liang. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 7(6):643–658, December 1999. ISSN 1063-6706.
- [50] J. M. Mendel and R. I. B. John. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, Apr 2002. ISSN 1063-6706.
- [51] Qilian Liang and J. M. Mendel. Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy Systems*, 8(5):535–550, 2000.
- [52] C. F. Juang, R. B. Huang, and Y. Y. Lin. A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing. *IEEE Transactions on Fuzzy Systems*, 17(5):1092–1105, Oct 2009. ISSN 1063-6706.
- [53] Ching-Hung Lee, Tzu-Wei Hu, Chung-Ta Lee, and Yu-Chia Lee. A recurrent interval type-2 fuzzy neural network with asymmetric membership functions for nonlinear system identification. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pages 1496–1502, June 2008.
- [54] Z. Deng, K. S. Choi, L. Cao, and S. Wang. T2fela: Type-2 fuzzy extreme learning algorithm for fast training of interval type-2 tsf fuzzy logic system. *IEEE Transactions on Neural Networks and Learning Systems*, 25(4): 664–676, April 2014. ISSN 2162-237X.

- [55] Juan R. Castro, Oscar Castillo, Patricia Melin, and Antonio Rodriguez-Daz. A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. *Information Sciences*, 179(13):2175 – 2193, 2009. ISSN 0020-0255. Special Section on High Order Fuzzy Sets.
- [56] A. K. Das, Nguyen Anh, Sundaram Suresh, and N. Srikanth. An interval type-2 fuzzy inference system and its meta-cognitive learning algorithm. *Evolving Systems*, 7(2):95–105, 2016. ISSN 1868-6486.
- [57] A. K. Das, K. Subramanian, and S. Sundaram. An evolving interval type-2 neurofuzzy inference system and its metacognitive sequential learning algorithm. *IEEE Transactions on Fuzzy Systems*, 23(6):2080–2093, 2015.
- [58] Evolving type-2 fuzzy classifier. *IEEE Transactions on Fuzzy Systems*, 24(3):547 – 589, 2015. ISSN 1941-0034.
- [59] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1):116–132, Jan 1985. ISSN 0018-9472.
- [60] E. H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Electrical Engineers, Proceedings of the Institution of*, 121(12):1585–1588, December 1974. ISSN 0020-3270.
- [61] C. F. Juang and Y. W. Tsao. A type-2 self-organizing neural fuzzy system and its fpga implementation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(6):1537–1548, Dec 2008. ISSN 1083-4419.
- [62] C. F. Juang and C. Y. Chen. An interval type-2 neural fuzzy chip with on-chip incremental learning ability for time-varying data sequence prediction and system control. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):216–228, Jan 2014. ISSN 2162-237X.

- [63] Rafał Scherer. *Ensembles of the Mamdani Fuzzy Systems*, pages 51–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-30604-4.
- [64] C. F. Juang, R. B. Huang, and W. Y. Cheng. An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems. *IEEE Transactions on Fuzzy Systems*, 18(4):686–699, Aug 2010. ISSN 1063-6706.
- [65] E. Lughofer and P. Angelov. Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Applied Soft Computing*, 11(2):2057 – 2068, 2011. ISSN 1568-4946. The Impact of Soft Computing for the Progress of Artificial Intelligence.
- [66] P. P. Angelov and D. P. 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.
- [67] K. Subramanian and S. Suresh. A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system. *Applied Soft Computing*, 12(11): 3603 – 3614, 2012. ISSN 1568-4946.
- [68] J. Kim and N. Kasabov. Hyfis: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. *Neural Networks*, 12(9): 1301 – 1319, 1999. ISSN 0893-6080.
- [69] W. L. Tung and C. Quek. A mamdani-takagi-sugeno based linguistic neural-fuzzy inference system for improved interpretability-accuracy representation. In *2009 IEEE International Conference on Fuzzy Systems*, pages 367–372, Aug 2009.
- [70] W. L. Ho, W. L. Tung, and C. Quek. An evolving mamdani-takagi-sugeno based neural-fuzzy inference system with improved interpretability-accuracy. In *International Conference on Fuzzy Systems*, pages 1–8, July 2010.

- [71] J. S. R. Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685, 1993.
- [72] Chia-Feng Juang and Chin-Teng Lin. An online self-constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, 6(1):12–32, 1998.
- [73] Sequential adaptive fuzzy inference system (safis) for nonlinear system identification and prediction. *Fuzzy Sets and Systems*, 157(9):1260 – 1275, 2006. ISSN 0165-0114.
- [74] Shiqian Wu and Meng Joo Er. Dynamic fuzzy neural networks-a novel approach to function approximation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 30(2):358–364, Apr 2000. ISSN 1083-4419.
- [75] C. J. Lin, C. H. Chen, and C. T. Lin. Efficient self-evolving evolutionary learning for neurofuzzy inference systems. *IEEE Transactions on Fuzzy Systems*, 16(6):1476–1490, Dec 2008. ISSN 1063-6706.
- [76] Detlef Nauck, Frank Klawonn, and Rudolf Kruse. *Foundations of Neuro-Fuzzy Systems*. John Wiley & Sons, Inc., New York, NY, USA, 1997. ISBN 0471971510.
- [77] Bartosz A. Nowak and Robert K. Nowicki. *Learning in Rough-Neuro-Fuzzy System for Data with Missing Values*, pages 501–510. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-31464-3.
- [78] E. D. Lughofer. Flexfis: A robust incremental learning approach for evolving takagi x2013sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 16(6):1393–1410, Dec 2008. ISSN 1063-6706.
- [79] F. Liu and M. J. Er. Design for self-organizing fuzzy neural network with extended kalman filter. In *IEEE ICCA 2010*, pages 423–427, June 2010.

- [80] H. J. Rong, G. B. Huang, N. Sundararajan, and P. Saratchandran. Online sequential fuzzy extreme learning machine for function approximation and classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(4):1067–1072, 2009.
- [81] W. B. Zhang and H. B. Ji. Fuzzy extreme learning machine for classification. *Electronics Letters*, 49(7):448–450, March 2013. ISSN 0013-5194.
- [82] C. T. Lin and C. S. G. Lee. Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, 40(12):1320–1336, Dec 1991. ISSN 0018-9340.
- [83] Detlef Nauck and Rudolf Kruse. Neuro-fuzzy systems for function approximation. *Fuzzy Sets and Systems*, 101(2):261 – 271, 1999. ISSN 0165-0114. Analytical and Structural Considerations in Fuzzy Modeling.
- [84] K. J. Park, S. K. Oh, and W. Pedrycz. Design of interval type-2 fuzzy neural networks and their optimization using real-coded genetic algorithms. In *2009 IEEE International Conference on Fuzzy Systems*, pages 2013–2018, 2009.
- [85] Ajith Abraham and Baikunth Nath. Evolutionary design of neuro-fuzzy systems – a generic framework. In *IN PROCEEDINGS OF THE 4 TH JAPAN – AUSTRALIA JOINT WORKSHOP ON INTELLIGENT AND EVOLUTIONARY SYSTEMS*, pages 106–113, 2000.
- [86] Zeke SH Chan and Nikola Kasabov. Evolutionary computation for on-line and off-line parameter tuning of evolving fuzzy neural networks. *International Journal of Computational Intelligence and Applications*, 4(03): 309–319, 2004.
- [87] F. Liu. Design for self-organizing fuzzy neural networks based on evolutionary programming. In *2010 Second International Conference on Computer Modeling and Simulation*, volume 2, pages 344–348, Jan 2010.

- [88] R. H. Abiyev and O. Kaynak. Type 2 fuzzy neural structure for identification and control of time-varying plants. *IEEE Transactions on Industrial Electronics*, 57(12):4147–4159, 2010.
- [89] Edwin Lughofer. *Evolving Fuzzy Systems - Methodologies, Advanced Concepts and Applications*. Springer-Verlag Berlin Heidelberg, 2011. ISBN 978-3-642-18087-3.
- [90] K. Subramanian, R. Savitha, and S. Suresh. Complex-valued neuro-fuzzy inference system for wind prediction. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2012.
- [91] An incremental type-2 meta-cognitive extreme learning machine. *IEEE Transactions on Cybernetics*, 47(2):339 – 353, 2016. ISSN 2168-2275.
- [92] Kartick Subramanian, Ankit Kumar Das, Suresh Sundaram, and Savitha Ramasamy. A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm. *Evolving Systems*, 5(4):219–230, Dec 2014. ISSN 1868-6486.
- [93] N. Kasabov. Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(6):902–918, Dec 2001. ISSN 1083-4419.
- [94] Panfis: A novel incremental learning machine. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):55 – 68, 2014. ISSN 2162-2388.
- [95] Genefis: Toward an effective localist network. *IEEE Transactions on Fuzzy Systems*, 22(23):547 – 562, 2014. ISSN 1941-0034.
- [96] Gang Leng, Girijesh Prasad, and T.M. McGinnity. An online algorithm for creating self-organizing fuzzy neural networks. 17:1477–93, 01 2005.
- [97] C. F. Juang and K. J. Juang. Reduced interval type-2 neural fuzzy system using weighted bound-set boundary operation for computation speedup

- and chip implementation. *IEEE Transactions on Fuzzy Systems*, 21(3):477–491, 2013.
- [98] Y. Y. Lin, J. Y. Chang, N. R. Pal, and C. T. Lin. A mutually recurrent interval type-2 neural fuzzy system (mrit2nfs) with self-evolving structure and parameters. *IEEE Transactions on Fuzzy Systems*, 21(3):492–509, June 2013. ISSN 1063-6706.
- [99] Y. Y. Lin, J. Y. Chang, and C. T. Lin. A tsk-type-based self-evolving compensatory interval type-2 fuzzy neural network (tscit2fnn) and its applications. *IEEE Transactions on Industrial Electronics*, 61(1):447–459, Jan 2014. ISSN 0278-0046.
- [100] M. Pratama, Jie Lu, and Guangquan Zhang. An incremental interval type-2 neural fuzzy classifier. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, Aug 2015.
- [101] A. Cai, C. Quek, and D. L. Maskell. Type-2 ga-tsk fuzzy neural network. In *2007 IEEE Congress on Evolutionary Computation*, pages 1578–1585, Sept 2007.
- [102] A. K. Das, S. Sundaram, and N. Sundararajan. A self-regulated interval type-2 neuro-fuzzy inference system for handling nonstationarities in eeg signals for bci. *IEEE Transactions on Fuzzy Systems*, 24(6):1565–1577, Dec 2016.
- [103] J. M. Mendel. Type-2 fuzzy sets and systems: An overview [corrected reprint]. *IEEE Computational Intelligence Magazine*, 2(2):20–29, 2007.
- [104] J. M. Mendel, R. I. John, and F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–821, Dec 2006. ISSN 1063-6706.

- [105] J. Y. Chang, Y. Y. Lin, M. F. Han, and C. T. Lin. A functional-link based interval type-2 compensatory fuzzy neural network for nonlinear system modeling. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 939–943, June 2011.
- [106] G. D. Wu and P. H. Huang. A vectorization-optimization-method-based type-2 fuzzy neural network for noisy data classification. *IEEE Transactions on Fuzzy Systems*, 21(1):1–15, Feb 2013. ISSN 1063-6706.
- [107] Nilesh N. Karnik and Jerry M. Mendel. Centroid of a type-2 fuzzy set. *Information Sciences*, 132(1):195 – 220, 2001. ISSN 0020-0255.
- [108] Maowen Nie and Woei Wan Tan. Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pages 1425–1432, June 2008.
- [109] K. Subramanian, S. Suresh, and N. Sundararajan. A metacognitive neuro-fuzzy inference system (mcfis) for sequential classification problems. *IEEE Transactions on Fuzzy Systems*, 21(6):1080–1095, Dec 2013. ISSN 1063-6706.
- [110] R Scott Crowder. Predicting the mackey-glass timeseries with cascade-correlation learning. pages 117–123, 01 1990.
- [111] P. Angelov and D. Filev. Simpl-lets: a simplified method for learning evolving takagi-sugeno fuzzy models. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pages 1068–1073, 2005.
- [112] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759.
- [113] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern Recognition*, 40(3):863 – 874, 2007. ISSN 0031-3203.

- [114] Data mining and wind power prediction: A literature review. *Renewable Energy*, 46(Supplement C):241 – 247, 2012. ISSN 0960-1481.
- [115] J. C. Patra, R. N. Pal, R. Baliarsingh, and G. Panda. Nonlinear channel equalization for qam signal constellation using artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(2):262–271, Apr 1999. ISSN 1083-4419.
- [116] S. Suresh, R. Savitha, and N. Sundararajan. A sequential learning algorithm for complex-valued self-regulating resource allocation network-csran. *IEEE Transactions on Neural Networks*, 22(7):1061–1072, 2011.
- [117] K. Subramanian, R. Savitha, and S. Suresh. A metacognitive complex-valued interval type-2 fuzzy inference system. *IEEE Transactions on Neural Networks and Learning Systems*, 25(9):1659–1672, 2014.
- [118] K. Subramanian, S. Suresh, and R. Venkatesh Babu. Meta-cognitive neuro-fuzzy inference system for human emotion recognition. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2012.
- [119] P. Angelov and X. Zhou. Evolving fuzzy systems from data streams in real-time. In *2006 International Symposium on Evolving Fuzzy Systems*, pages 29–35, Sept 2006.
- [120] Meta-cognitive rbf network and its projection based learning algorithm for classification problems. *Applied Soft Computing*, 13(1):654 – 666, 2013. ISSN 1568-4946.
- [121] J. Waldron J.J. de Vries and V. Cunningham. Field tests of the new datawell dwr-g gps wave buoy. 44:50–55, 12 2003.
- [122] G. Jeans, I. Bellamy, J. J. de Vries, and P. van Weert. Sea trial of the new datawell gps directional waverider. In *Proceedings of the IEEE/OES Seventh Working Conference on Current Measurement Technology, 2003.*, pages 145–147, March 2003.

- [123] A. Khosravi, S. Nahavandi, and D. Creighton. Prediction interval construction and optimization for adaptive neurofuzzy inference systems. *IEEE Transactions on Fuzzy Systems*, 19(5):983–988, 2011.
- [124] M. Esposito and D. Maisto. *A Framework for Verification of Fuzzy Rule Bases Representing Clinical Guidelines*, pages 545–557. Springer New York, New York, NY, 2013. ISBN 978-1-4614-3535-8.
- [125] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, Mar 1990. ISSN 1045-9227.