IMPROVING UNDERSTANDABILITY AND UNCERTAINTY MODELING OF DATA USING FUZZY LOGIC SYSTEMS

Dumidu S. Wijayasekara
Virginia Commonwealth University, wijayasekards@vcu.edu

Follow this and additional works at: http://scholarscompass.vcu.edu/etd
Part of the Computer Engineering Commons

© The Author

Downloaded from
http://scholarscompass.vcu.edu/etd/4126

This Dissertation is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.
IMPROVING UNDERSTANDABILITY AND UNCERTAINTY MODELING
OF DATA USING FUZZY LOGIC SYSTEMS

A Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at Virginia Commonwealth University

by

DUMIDU SHANIKA WIJAYASEKARA
Bachelor of Science, University of Peradeniya, Sri Lanka, 2009
Master of Science, University of Idaho, USA, 2014

Director: Milos Manic
Professor, Department of Computer Science, School of Engineering

Virginia Commonwealth University
Richmond, Virginia
May 2016
Acknowledgements

First, I would like to acknowledge and thank my advisor and mentor Prof. Milos Manic for his help, support and guidance. Second, I would like to thank Prof. Vojislav Kecman, Dr. Preetam Ghosh, Dr. Supathorn Phongikaroon, and Dr. Craig Rieger for agreeing to join my dissertation committee and for providing guidance and valuable feedback that made the dissertation better. I would also like to thank my colleagues for their immense help and support. Finally, I would like to acknowledge the support given to me by the Idaho National Laboratory.

I also wish to thank all my friends and family for their assistance and support during the academic process.


**TABLE OF CONTENTS**

Acknowledgements........................................................................................................... i

List of Figures .................................................................................................................... vi

List of Tables ..................................................................................................................... ix

Abstract ............................................................................................................................. xi

Chapter 1  Introduction..................................................................................................... 1

1.1  Motivation ................................................................................................................. 3

1.2  Objectives of the Dissertation ................................................................................. 3

1.3  Contributions of the Dissertation ............................................................................ 4

1.4  Organization of the Dissertation .............................................................................. 6

Chapter 2  Fuzzy Logic Sets and Systems...................................................................... 8

2.1  Type-1 Fuzzy Logic (T1 FL) ................................................................................... 8

2.1.1  Type-1 Fuzzy Sets (T1 FSs) .............................................................................. 9

2.1.2  Type-1 Fuzzy Set Theoretic Operations ......................................................... 12

2.1.3  Type-1 Fuzzy Logic Systems (T1 FLS) ............................................................ 14

2.2  Type-2 Fuzzy Logic (T2 FL) .................................................................................. 17

2.2.1  General Type-2 Fuzzy Sets (GT2 FSs) ........................................................... 17

2.2.2  Representations of GT2 FSs ........................................................................... 19

2.2.3  Set Theoretic Operations on GT2 FSs ........................................................... 20

2.2.4  General Type-2 Fuzzy Logic Systems (GT2 FLSs) ....................................... 22
2.3 Interval Type-2 Fuzzy Logic (IT2 FL) ........................................................................ 24
  2.3.1 Interval Type-2 Fuzzy Sets (IT2 FSs) .............................................................. 24
  2.3.2 Set Theoretic Operations on IT2 FSs ............................................................ 25
  2.3.3 Interval Type-2 Fuzzy Logic Systems (IT2 FLSs) ............................................. 26

2.4 Conclusions ......................................................................................................... 28

Chapter 3 Generating Fuzzy Sets for Understandability ............................................ 30
  3.1 Understandability of Fuzzy Systems ................................................................. 31
  3.2 Generating Fuzzy Membership Functions ......................................................... 32
  3.3 Generating Fuzzy Membership Functions for Understandability ..................... 33
    3.3.1 Understandability Metrics ............................................................................. 34
    3.3.2 Related Work ............................................................................................... 39
    3.3.3 Presented Data Driven Method for Generating Membership Functions . 40
    3.3.4 Convergence of the Presented Method ....................................................... 46

3.4 Experimental Results .......................................................................................... 47
    3.4.1 Experimental Setup ...................................................................................... 48
    3.4.2 Experimental Results .................................................................................... 53

3.5 Conclusions ......................................................................................................... 54

Chapter 4 Linguistic Summarization of Data Using Fuzzy Logic ............................... 55
  4.1 Linguistic Summarization of Data ...................................................................... 56
    4.1.1 Yager-Type Linguistic Summarization ......................................................... 60
6.1.2 Data Description ................................................................. 112
6.1.3 Anomaly Detection and Graphical User Interface .................. 114
6.1.4 Experimental Results .......................................................... 117
6.2 Fuzzy Logic for Uncertainty Modeling in Real-World Systems .......... 124
  6.2.1 Haptics for Robot Teleoperation ........................................ 124
  6.2.2 System Description ............................................................. 126
  6.2.3 Experimental Results .......................................................... 132
6.3 Conclusions ............................................................................. 141
Chapter 7 Conclusion And Future Work ........................................... 142
  7.1 Final Conclusions .................................................................. 142
  7.2 Future Work ........................................................................... 144
Appendix A List of Publications by the Author ..................................... 146
  Journal Publications ..................................................................... 147
  Peer-Reviewed Conference Publications ........................................ 149
References ...................................................................................... 158
Vitae ............................................................................................... 187
LIST OF FIGURES

Figure 1. Typical T1 fuzzy MF (a) triangular MF, (b) trapezoidal MF, (c) Gaussian MF, (d) generalized Bell MF ................................................................. 11

Figure 2. Type-1 Fuzzy Logic System ....................................................................................... 15

Figure 3. General Type-2 Fuzzy Logic System ........................................................................ 21

Figure 4. Fuzzy Membership Functions (MFs) and their parameters demonstrated using trapezoidal MFs ............................................................................. 34

Figure 5. Generated prototypes for the benchmark bivariate dataset at each step ((a)- (c)) and the final generated MFs (d) ...................................................................................... 50

Figure 6. Initial prototypes for (a) uniformly distributed data, (c) normally distributed data, and (e) right skewed data, along with generated MF for, (b) uniformly distributed data, (d) normally distributed data, and (f) right skewed data. ....... 51

Figure 7. Initial prototypes for real world zone temperature data - (a), (c) and (e), along with generated MF - (b), (d) and, (f) .............................................................. 52

Figure 8. Fuzzy sets used for linguistic summarization (a) input data dimension fuzzy sets (b) quantifier fuzzy sets for Yager type rules ................................................................. 59

Figure 9. Sigmoid function used to calculate the degree of sufficient coverage ..................... 65

Figure 10. Run time comparison of Yager type summaries (a) IF-THEN type summaries (b) using different sized datasets ................................................................. 76

Figure 11. Quality degradation of generated “Yager” type summaries at different noise levels for each dataset (a) Auto MPG (b) Blood Transfusion (c) Pima Indian Diabetes, and (d) Wine Quality ................................................................. 81
Figure 12. Quality degradation of generated “IF-THEN” type summaries at different noise levels for each dataset (a) Auto MPG (b) Blood Transfusion (c) Pima Indian Diabetes, and (d) Wine Quality ................................................................. 82

Figure 13. Type-1 fuzzy set and the induced shadowed set depicting the three regions of the shadowed set (Core, Shadow, and Exclusion) ........................................ 90

Figure 14. Generating ST2 FS (a) Secondary membership function of a GT2 FS and its segmentation using two $\alpha$-planes (b) the optimization function $V(\tilde{A})$ ....... 93

Figure 15. Generating ST2 FS (a) GT2 FS $\tilde{A}$ (b) the ST2 FS induced by $\tilde{A}$, $\tilde{A}$ ............. 96

Figure 16. Set theoretic operations on ST2 FSs (a) two initial ST2 FSs (b) intersection operation (c) union operation (d) complement operation........................................ 97

Figure 17. Centroid of a type reduced ST2 FS ................................................................. 99

Figure 18. Fuzzy inference (a) architecture of a Mamdani type ST2 FLS (b) Parallel processing inner and outer boundaries of the ST2 FLS ......................... 100

Figure 19. GT2 FSs of the GT2 FLS. (a) inputs, (b) outputs.................................................. 103

Figure 20. ST2 FSs directly induced by the GT2 FSs, (a) inputs, (b) outputs....................... 105

Figure 21. Output surfaces of each FLS (a) ST2 FLS (b) the original GT2 FLS .............. 105

Figure 22. IT2 FSs generated using the FOU of the GT2 FSs, (a) inputs, (b) outputs....... 106

Figure 23. Squared errors of the IT2 FLS (a) and the ST2 FLS (b) compared to the GT2 FLS (Note the different scale in the vertical axis) ............................. 107

Figure 24. Typical operational characteristics of a selected office room for a 48 hour period........................................................................................................... 113

Figure 25. Implemented GUI with (a) building view, (b) floor view, and (c) data view .... 115
Figure 26. Floor view depicting (a) normal behavior (b) abnormal behavior using temperature as color, and (c) abnormal behavior utilizing anomaly level as color.

Figure 27. Fuzzy sets and linguistic labels used for the time attribute.

Figure 28. Abnormal building behavior tested (a) – (c) sensor based anomalies, (d) – (f) physical anomalies.

Figure 29. Virtual force field based force-feedback calculation (a) virtual force field representation for object $K$ in axis $i$, (b) force-field overlap calculation in axis $i$.

Figure 30. Overall dynamic force-feedback generation method.

Figure 31. Hardware implementation for testing (a) schematic of the manipulator, (b) the Novint Falcon force-feedback device.

Figure 32. GT2 Fuzzy Sets for (a) distance, (b) velocity and (c) force-field spread.

Figure 33. IT2 Fuzzy Sets for (a) distance, (b) velocity and (c) force-field spread.

Figure 34. ST2 Fuzzy Sets for (a) distance, (b) velocity and (c) force-field spread.

Figure 35. Performance of each type of FLS at different noise levels.

Figure 36. Experimental task for testing different methods for force-feedback generation.
LIST OF TABLES

Table 1. Preset constraints for understandability metrics .......................................................... 49
Table 2. Similarity of the generated FLS at different noise levels ............................................ 50
Table 3. Benchmark datasets used for evaluation ........................................................................ 73
Table 4. Comparison between the summaries generated by the Exhaustive method
and the SOM method for Auto MPG dataset ............................................................................. 78
Table 5. Comparison between the summaries generated by the Exhaustive method
and the SOM method for Blood Transfusion dataset ................................................................. 78
Table 6. Comparison between the summaries generated by the Exhaustive method
and the SOM method for Pima Indian Diabetes dataset ............................................................. 79
Table 7. Comparison between the summaries generated by the Exhaustive method
and the SOM method for Wine Quality dataset ......................................................................... 79
Table 8. Fuzzy rule base used for comparison ............................................................................. 103
Table 9. Comparison of the uncertainty modeling and run-time of ST2 FLS and IT2
FLS with GT2 FLS ................................................................................................................. 107
Table 10. List of extracted attributed and their scope ................................................................. 113
Table 11. Generated Yager type summaries for the dataset ......................................................... 118
Table 12. Generated IF-THEN type summaries for the dataset ................................................. 119
Table 13. Tested anomalous behavior scenarios ......................................................................... 120
Table 14. Generated IF-THEN type summaries for each case and the detected times .......... 120
Table 15. Fuzzy rule-base used to dynamically generate the force-field in each axis ............ 133
Table 16. Average task completion times for each method at different noise levels ........... 140
Table 17. Average accuracy compared to the furthest distance path for each method

at different noise levels........................................................................................................ 140
Abstract

IMPROVING UNDERSTANDABILITY AND UNCERTAINTY MODELING OF DATA USING FUZZY LOGIC SYSTEMS

By Dumidu Shanika Wijayasekara

A Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at Virginia Commonwealth University

Virginia Commonwealth University, 2016

Director: Milos Manic
Professor, Department of Computer Science, School of Engineering

The need for automation, optimality and efficiency has made modern day control and monitoring systems extremely complex and data abundant. However, the complexity of the systems and the abundance of raw data has reduced the understandability and interpretability of data which results in a reduced state awareness of the system. Furthermore, different levels of uncertainty introduced by sensors and actuators make interpreting and accurately manipulating systems difficult. Classical mathematical methods lack the capability to capture human knowledge and increase understandability while modeling such uncertainty.

Fuzzy Logic has been shown to alleviate both these problems by introducing logic based on vague terms that rely on human understandable terms. The use of linguistic terms and simple consequential rules increase the understandability of system behavior as well as data. Use of vague terms and modeling data from non-discrete prototypes enables modeling of uncertainty.
However, due to recent trends, the primary research of fuzzy logic have been diverged from the basic concept of understandability. Furthermore, high computational costs to achieve robust uncertainty modeling have led to restricted use of such fuzzy systems in real-world applications. Thus, the goal of this dissertation is to present algorithms and techniques that improve understandability and uncertainty modeling using Fuzzy Logic Systems.

In order to achieve this goal, this dissertation presents the following major contributions: 1) a novel methodology for generating Fuzzy Membership Functions based on understandability, 2) Linguistic Summarization of data using if-then type consequential rules, and 3) novel Shadowed Type-2 Fuzzy Logic Systems for uncertainty modeling. Finally, these presented techniques are applied to real world systems and data to exemplify their relevance and usage.
CHAPTER 1

INTRODUCTION

Modern day control and monitoring systems have become extremely complex and data abundant as the constraints for efficiency and optimality are continuously being increased in a highly competitive environment. Furthermore, the increasing need for automation of control systems increases the complexity of the available data even more. Such complexity and large amount of raw data leads to reduced understandability and interpretability of the overall system, and therefore a lowered state awareness [1]-[3]. Furthermore, different levels of uncertainty are introduced by different sensors and actuators that makes interpretation and manipulation of systems difficult in dynamic environments. While classical mathematical models lack the capability of increasing understandability and handling certain types of uncertainty, Fuzzy Logic has been shown to be able to model such parameters effectively [4]-[6].

Fuzzy Logic was first introduced by Lotfi Zadeh in 1965 [1] and is a well-documented technique that has been successfully used in various applications since its inception over 5 decades ago [4]-[9]. In classical Boolean Logic based operations, an object either belongs to a set or does not, i.e. they are crisp. In Fuzzy Logic however, the degree of belonging, known as the membership, can range from any value between 0 and 1. Thus, it is argued that vague concepts of perception cannot be modelled by classical mathematics and Fuzzy Logic is more suited to model such perceptions (e.g. size, comfort, safety, etc.) and vagueness [4], [8]. Furthermore, this “soft” degree of belonging enable modeling of the concept of overlapping
prototypes [10]. Thus, human understandable linguistic terms which are inherently vague and objective, can be modelled using Fuzzy Logic [4], [11].

Fuzzy Logic Systems (FLSs) take advantage of the capabilities of Fuzzy Logic and computes based on a human understandable rule-base [12], [13]. FLSs are a proven methodology for control, classification, data-mining and various other applications [11], [14], [15].

The capability of FLSs to use human like linguistic terms to represent vague terms and relationships leads to bridging the gap between machines and humans through techniques such as computing with words [4], [6], [8]. Furthermore, the use of such terms and human interpretable rules for modeling complex systems, yield highly understandable and interpretable, white/grey box systems [16]. The vague terms and membership grades are also capable of modeling uncertainties inherent to data and systems, which leads to accurate and capable systems even in the presence of uncertainties [6], [10].

The primary advantages of FLSs are three fold: 1) use of linguistic terms [4], [16], 2) understandability and interpretability [11], [16], [17], [18], and 3) the capability of modeling uncertainty [6], [8], [10], [16], [19]. The understandability, uncertainty modeling and other capabilities of FLSs have also led to their prolific use in real-world control systems, data mining systems and consumer electronics [8].

This dissertation presents novel methodologies for improving understandability and uncertainty modeling of Fuzzy Logic Systems. For improved understandability, a data driven Fuzzy Membership Function (MF) generation methodology and a linguistic summarization
framework is presented. For improved uncertainty modeling, novel fuzzy inference system called Shadowed Type-2 Fuzzy Logic System (ST2 FLS) is presented.

1.1 Motivation

Two of the primary advantages of FLSs are improving understandability and uncertainty modeling of systems and data [6], [8], [20]. However, recent bias towards efficiency and better performance has resulted in many research projects leading away from the fundamental understandability of FLSs and focusing more on the performance aspect of FLSs [11]. Furthermore, while recent advancements in Type-2 Fuzzy Logic have improved the uncertainty modeling capability of FLSs, the high computational complexity associated with such systems has led to limited use of Type-2 FLSs in real-world systems [6], [8], [19]. Thus, elaborating and enhancing understandability and uncertainty modeling of fuzzy sets and systems is an important and relevant research task.

1.2 Objectives of the Dissertation

The objective of this dissertation is to develop and test algorithms and techniques to improve understandability and uncertainty modeling of data and systems by means of fuzzy sets and systems. In this context, understandability and uncertainty modeling can be further clarified as follows.

Understandability of a fuzzy rule, which is a combination of linguistic terms, can be loosely defined as an ability to read and understand all information described by the fuzzy rule
However, this definition further depends on the notion that what is understood by the reader is the same knowledge that is contained in the data [14], [21], [23]. To achieve this, the linguistic terms should be consistent with human intuition and in-line with expert opinions as well as the actual data that is modelled by them [11], [14], [23]. Thus, understandability of systems and data that take advantage of capabilities of FLSs depend on fuzzy sets, granularity of the fuzzy sets, and linguistic terms as well as the quantity and quality of the fuzzy rules [11], [22]. Furthermore, this definition encompasses the notions of interpretability, transparency and complexity of fuzzy sets and systems [14], [22].

Uncertainty modeling is an important and necessary aspect of system modeling as well as knowledge extraction applications [6], [8], [19]. For well-defined uncertainties, known as stochastic or aleatory uncertainties, classical probabilistic techniques for uncertainty modeling are sufficient [5], [6], [20], [24]. However, when knowledge about the uncertainties present in the system is limited (epistemic uncertainties), fuzzy systems have been shown to be powerful uncertainty modeling techniques [6], [24]-[27]. Furthermore, classical probabilistic techniques lack the capability to model epistemic uncertainties [6], [20], [25]. Thus, in this dissertation, epistemic uncertainties and modeling epistemic uncertainties will be discussed. In typical engineering applications, epistemic uncertainties are manifested in terms of noise in measurements [6], [28].

1.3 Contributions of the Dissertation

In order to achieve the objectives outlined above, three primary contributions are presented in this dissertation.
First, a novel, data driven methodology for generating Fuzzy Membership Functions that increases overall understandability of Fuzzy Logic Systems is presented. The presented methodology does not rely on an iterative clustering or evolutionary algorithms and is capable of satisfying the most commonly used understandability metrics. Furthermore, two new understandability metrics are also presented that further increases the understandability of the generated Membership Functions.

Second, a methodology for fast and robust Linguistic Summarization of data using IF-THEN rules that utilizes Self-Organizing Maps is presented. The presented method is also capable of generating other types of linguistic summaries. Furthermore, several quality metrics for Linguistic Summaries are also presented.

Third, a novel Fuzzy Logic System that reduces computation time while maintaining the uncertainty modeling capabilities of General Type-2 Fuzzy Logic Systems, known as Shadowed Type-2 Fuzzy Logic Systems are presented. The presented Shadowed Type-2 Fuzzy Logic Systems are directly derived from General Type-2 Fuzzy Logic Systems and therefore maintain high levels of uncertainty modeling capability. However, representation of Shadowed Type-2 Fuzzy Sets enable faster computation times compared to General Type-2 Fuzzy Logic Systems.

The algorithms and methodologies presented in this dissertation are also tested on real-world systems and data sets verifying their relevance. Furthermore, the experiments demonstrate the usability of the presented methodologies in real-world scenarios.
1.4 Organization of the Dissertation

Chapter 2 provides an overview of different types of prevalent FLS. This chapter starts by defining the basic theory behind Fuzzy Logic and other relevant terms related to FLS and follows with a general overview of Type-1, General Type-2 and Interval type-2 FLS. Finally, it is concluded by reviewing the different types of FLS, their advantages and where each type of FLS can be used.

Chapter 3, first, introduces concept of understandability in relationship to Fuzzy Membership Functions (MFs), and its importance in usability of FLSs. The generation of MFs for FLSs and typically used methods for generation of MF are then discussed with advantages and disadvantages of each method being highlighted. Next, a novel method for generating MF for understandability is presented, along with understandability metrics used to measure the quality of the generated MFs. Chapter 3 is concluded by presenting experimental results on benchmark and real-world datasets, and identifying the importance of the presented method and its consequences. Chapter 4 presents Linguistic Summarization (LS) of data using IF-THEN rules for increased understandability. Various LS techniques are presented that can be useful for data-mining and data pre-exploration. A novel methodology for ranking terms is presented along with experimental results. Then, a LS methodology using Self-Organizing Maps (SOM) is presented that is insusceptible to noise in the data. Chapter 4 is concluded by discussing the importance of LS for data-mining and data pre-exploration and possible improvements to the presented methods.

Chapter 5 presents the novel Shadowed Type-2 (ST2) FLSs. This chapter first details the advantages and shortcomings of both General Type-2 (GT2) and Interval Type-2 (IT2) FLS and exemplifies the importance ST2 FLSs for accurate and computationally efficient
uncertainty modeling. Then, the ST2 Fuzzy Sets are introduced and the generation of ST2 FLSs and computations of ST2 FLSs are detailed. The computational advantages of ST2 FLSs along with the gained uncertainty modeling capability of the presented ST2 FLSs is then discussed through experimental results. Finally, this is concluded by discussing the implications of the experimental results and the relative benefits of using the presented ST2 FLS for real-world problems.

Chapter 6 provides several examples of improved understandability and uncertainty modeling used in real-world systems. First, an architecture for understanding overall system behavior as well as anomalous states in buildings that utilizes understandable Fuzzy MFs and LS methods presented in this dissertation is described. Second, a force feedback for robot teleoperation mechanism that relies on the uncertainty modeling capabilities of the presented ST2 FLSs is described. Experimental results for both real-world examples are presented in this chapter. Chapter 6 is concluded by discussing the advantages and disadvantages of the presented methods in real-world scenarios as well as possible future improvements. Chapter 7 provides overall conclusions and suggests directions for future work.
CHAPTER 2

FUZZY LOGIC SETS AND SYSTEMS

Fuzzy Logic (FL) was first conceived by Lotfi Zadeh in 1965 [1]. The primary driver for the introduction of FL was to explain complex systems in an understandable manner [1], [2], [16]. Furthermore, modeling complex phenomena that is difficult or sub-optimal to be modelled by classical mathematics is also an objective of FL [18], [29]. FL can be viewed as a system that provides a methodology for modeling and calculating human like imprecision and reasoning [2], [16], [26], [29], [30]. FL relies on Fuzzy set theory for representation of imprecise models and reasoning. Fuzzy set theory is similar to classical set theory but uses Fuzzy Sets (FSs) instead of classical sets [16], [26], [29]. In the five decades that passed since its introduction, FL has been applied in various fields and has undergone many algorithmic advances [6], [26], [29]. However, the basic theories and ideas introduced by Zadeh in [1], still holds. Thus, this chapter provides a background overview of FL and some basic set theoretic operations and inference mechanism that are relevant to the work presented in this dissertation. This chapter also provides a brief overview of Type-2 FL which was also presented by Zadeh in 1975 [18], as it is also relevant to the work presented in this dissertation.

2.1 Type-1 Fuzzy Logic (T1 FL)

Fuzzy Logic (FL) is analogous to Boolean logic. However, while Boolean logic uses classical crisp sets to model precise sets and reasoning, FL uses Fuzzy Sets (FSs) that models
imprecise sets and reasoning. Thus, FL relies of FSs and Fuzzy set theoretic operations for representing and modeling as well as computing [16].

2.1.1 Type-1 Fuzzy Sets (T1 FSs)

Classical Boolean logic expresses a crisp set $S$ in universe of disclosure (or domain) $X$ by either listing all its elements, or by providing a conditional property of members of $S$, or by specifying a binary characteristic function $\mu_S \in \{0, 1\}$. This binary characteristic function is also known as the Membership Function (MF) and can be expressed as:

$$\mu_S(x) = \begin{cases} 1, & \text{if } x \in S \\ 0, & \text{if } x \notin S \end{cases}, \forall x \in X \quad (2.1)$$

Thus, in crisp sets, the element $x$ can either completely belong to $S$ or not belong to $S$ at all. In contrast, fuzzy set theory assigns a membership degree according a degree of belonging to a certain set. Thus, typically, a fuzzy MF can take any arbitrary value between 0 and 1 (inclusive), i.e., $\mu_A \in [0, 1]$. The FS $A$ in the domain $X$ can be expressed as a set of ordered pairs of elements and their respective membership degrees [6]:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (2.2)$$

This representation, when described in the continuous domain, leads to the commonly used notation [6], [31], [32]:

$$A = \int_{x \in X} \frac{\mu_A(x)}{x} \quad (2.3)$$
where \( \int_{x \in X} \) symbolizes the collection of all points in the domain \( X \) \([6], [33]\).

A FS that can only have membership degrees in the interval \([0, 1]\) is known as a \textit{Normal} FS \([29]\). While, FS that may have any arbitrary membership degrees are used in some cases, \textit{Normal} FS are more widely used for understandability \([22], [34], [35]\). Thus, in this dissertation, all FS used will be \textit{Normal} FS. A given FS can be completely characterized by its respective membership function \([6], [30]\). As FS are typically used in real world applications where the domain is continuous, the MF are typically expressed as parameterized mathematical functions. The most commonly used MF (and the types of MF used in this dissertation) are triangular, trapezoidal, Gaussian, and generalized Bell curves, for which the Equations are given in \((2.4), (2.5), (2.6)\) and \((2.7)\), respectively. Figure 1 depicts examples of each type of MF \([6], [30]\).

\[
\mu_{A}(x) = \begin{cases} 
0, & \text{if } x < a \\
\frac{x-a}{c-a}, & \text{if } a \leq x \leq c \\
\frac{b-x}{b-c}, & \text{if } c < x \leq b \\
0, & \text{if } x > b 
\end{cases} 
\] (2.4)
Figure 1. Typical T1 fuzzy MF (a) triangular MF, (b) trapezoidal MF, (c) Gaussian MF, (d) generalized Bell MF

\[ \mu_A(x) = \begin{cases} 
0, & \text{if } x < a \\
\frac{x-a}{m-a}, & \text{if } a \leq x < m \\
1, & \text{if } m \leq x \leq n \\
\frac{b-x}{b-n}, & \text{if } n < x \leq b \\
0, & \text{if } x > b 
\end{cases} \]  

(2.5)

\[ \mu_A(x) = \exp\left(\frac{(x-c)^2}{2\sigma^2}\right) \]  

(2.6)
\[ \mu_A(x) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}} \quad (2.7) \]

2.1.2 Type-1 Fuzzy Set Theoretic Operations

Similar to classical set theory, FL also has set theoretic operations, which are analogous to classical set theoretic operations. The most basic of these operations are the operations that were introduced by Zadeh in [1]. These operations are containment, intersection, union and complement [1], [5], [30]. The fuzzy intersection and union operations are analogous to Boolean AND and Boolean OR operations respectively. The containment operation specifies the equivalent subset operation in classical set theory. FS \( A \) is contained in FS \( B \), if and only if \( \mu_A(x) \leq \mu_B(x) \) for all \( x \) in domain \( X \), which can be expressed as [1], [30]:

\[ A \subseteq B \iff \mu_A(x) \leq \mu_B(x) \quad \forall x \in X \quad (2.8) \]

The intersection operation of two FS \( A \) and \( B \), expressed as \( A \cap B \), can be represented in terms of their respective MF as follows:

\[ A \cap B = \mu_{A \cap B}(x) = \mu_A(x) \amalg \mu_B(x) \quad (2.9) \]

where, the operator \( \amalg \) denotes a fuzzy t-norm operation [6], [29].

Similarly, the union operation of two FS \( A \) and \( B \), is expressed as \( A \cup B \), and represented in terms of their respective MF as:

\[ A \cup B = \mu_{A \cup B}(x) = \mu_A(x) \uplus \mu_B(x) \quad (2.10) \]
where, the operator \( \prod \) denotes a fuzzy t-conorm or s-norm operation [6], [29].

The fuzzy t-norm and t-conorm operations can be any operation that satisfies the conditions of monotonicity, commutativity and associativity [36]. For a t-norm or t-conorm operation \( T \), monotonicity can be expressed as [36]:

\[
\begin{align*}
    b \leq d & \Rightarrow (T(a,b)) \leq (T(a,d)) \\
    \forall a, b, d & \in [0,1]
\end{align*}
\]  \hspace{1cm} (2.11)

Commutativity can be expressed as [36]:

\[
T(a, b) = T(b, a) \quad \forall a, b \in [0,1]
\]  \hspace{1cm} (2.12)

Associativity is expressed as [36]:

\[
T(a, T(b, d)) = T(T(a, b), d) \quad \forall a, b, d \in [0,1]
\]  \hspace{1cm} (2.13)

In addition to the above conditions, t-norm and t-conorm has to conform to the following boundary conditions, \( \forall a \in [0,1] \):

\[
a \prod 1 = a
\]  \hspace{1cm} (2.14)

\[
a \prod 0 = a
\]  \hspace{1cm} (2.15)

While many different variations of t-norm and t-conorm can be found in literature and each has its merit [29], the most widely used and the initially presented type of t-norm and t-conorm are min and max operators respectively [1], [6], [30]. Thus, the intersection and union operations can be expressed as:

\[
A \cap B = \min(\mu_A(x), \mu_B(x))
\]  \hspace{1cm} (2.16)

\[
A \cup B = \max(\mu_A(x), \mu_B(x))
\]  \hspace{1cm} (2.17)
All t-norm and t-conorm operations in this dissertation uses the *min* and *max* operators respectively unless otherwise specified. The complement of a FS $A$ can be expressed as:

$$\overline{A} = \mu_{\overline{A}}(x) = 1 - \mu_A(x) \quad (2.18)$$

### 2.1.3 Type-1 Fuzzy Logic Systems (T1 FLSs)

As mentioned, FL has been successfully applied in various different fields in various different forms since its inception [2], [5], [6]. However, one of the most prevalent and successful applications of FL is with Fuzzy Logic Systems (FLSs). In its most abstract form, a FLS can be seen as a non-linear system that maps an input vector into a scalar output [6]. The non-linear mapping is done using a rule base which is encoded using human interpretable linguistic rules. The use of such a rule base enables both encoding human knowledge into a complex system and easy interpretation of the system [6], [13], [26]. Typically, FS are used to encode input and output domains and the basic set theoretic operations discussed above are used to generate an output given an input vector. A FLS that uses only Type-1 FS is a known as a Type-1 FLS [6].

Two major types of FLSs are typically used in literature: the Mamdani type FLS [13] and Takagi-Sugeno-Kang (TSK) type FLS [12], [37]. The primary difference between the two types are that Mamdani type TFS uses FS to encode the outputs whereas the TSK type FLS
uses a functional mapping to encode outputs [19], [30]. The Mamdani type FLS will be further elaborated as it is the type that is mainly used in this dissertation.

A FLS consists of four major parts: 1) rule base, 2) fuzzification, 3) inference, and 4) defuzzification [6] (as shown in Figure 2). The rule base contains the human knowledge that drives the system. Fuzzification converts a crisp input vector into a set of membership degrees. Inference is the process that generates an output FS using the given set of rule and membership values. Finally, defuzzification produces a crisp output [6].

A rule in a Mamdani type FLS, which are also known as fuzzy implication rules, takes the form of IF-THEN rules [6], [30]. The portion between the IF statement and the THEN statement is known as the antecedent and the portion after the THEN statement is known as the consequent [6]. Thus, the $k^{th}$ rule, $R_k$ in a FLS with $K$ rules, $N$ input dimensions where $x_i \in X_1, \ldots, x_N \in X_N$ and one output dimension $y \in Y$ is represented as:

$$R_k: \text{IF } x_1 \text{ is } A_1^k \text{ AND } \ldots \text{ AND } x_n \text{ is } A_n^k \text{ THEN } y \text{ is } B^k$$

(2.19)

where $n \leq N$ is the number of antecedents of rule $R_k$, $A_i^k$ is the FS for the $i^{th}$ input dimension and $B^k$ is the FS of the output dimension. Thus, a rule consists of one or more
antecedents combined with AND clauses and a single consequent. It can be easily shown that any other combination of rules can be decomposed into either one or several of the rules that take the form shown in (2.19) [6].

The MF of the rule $R_k$ in (2.19), $\mu_{R_k}(\bar{x}, y)$ can be formulated as:

$$\mu_{R_k}(\bar{x}, y) = \mu_{A'_i}(x_i) \prod \mu_{A'_n}(x_n) \prod \mu_{B^i}(y)$$  \hspace{1cm} (2.20)

where, $\bar{x} = (x_1, x_2, \ldots, x_n)$ is the input vector to the rule $R_k$ and $\mu_{B_i}(y)$ is the FS for the domain of the output. This means that the output FS $B^k$ is clipped by the minimum of the antecedent membership degree [6]. For a FLS with $K$ rules, the final output FS $B(y)$ is calculated using:

$$B(y) = \prod_{i=1}^{K} \mu_{R_k}(\bar{x}, y)$$  \hspace{1cm} (2.21)

Equations (2.20) and (2.21) represent the inference stage of the FLS. The final defuzzification stage decomposes the output FS generated in the previous step into a crisp value. Many different methods exist in literature for this step as well [6], [29]. However, in this dissertation, the centroid defuzzification is used. Centroid defuzzification can be expressed using the membership degree of the output FS $\mu_B(y)$ as:

$$y(\bar{x}) = \frac{\sum_{i=1}^{M} y_i \cdot \mu_B(y_i)}{\sum_{i=1}^{M} \mu_B(y_i)}$$  \hspace{1cm} (2.22)

where, $M$ is a set of discretized samples in the output domain $Y$, and $y(\bar{x})$ is the final scalar output for the input vector $\bar{x}$. 

16
Thus, given an input vector the FLS is capable of utilizing the rule base to generate a scalar output.

2.2 Type-2 Fuzzy Logic (T2 FL)

Type-2 Fuzzy Logic (T2 FL) was first introduced by Zadeh in 1975 [18]. The reason for introducing T2 FL is that the uncertainty modeling capability of T1 FL may not be sufficient in most real-world scenarios [6], [10], [19], [38]. T2 FL transforms the representation of the membership grade in the interval of [0, 1] in T1 FL, in to a T1 MF of its own. In other words, the membership grade of a T2 FS is a T1 FS, rather than a crisp value in the case of T1 FL. There are two major types of T2 FL that are being used in the literature: General Type-2 Fuzzy Logic (GT2 FL) and Interval Type-2 Fuzzy Logic (IT2 FL). GT2 FL is T2 FL in its original form and IT2 FL is a special case of GT2 FL that is being used more frequently because of the high computational complexity of GT2 FL. Thus, each type of T2 FL has its own representation and operations. This section will briefly describe only the representations and operations of both GT2 and IT2 FL that are relevant to this dissertation.

2.2.1 General Type-2 Fuzzy Sets (GT2 FSs)

As mentioned, a GT2 FS is a FS where the membership degree is a FS by itself. Thus, a GT2 FS essentially has two membership degrees: one over the input (or output) domain, which is called the primary membership degree, and one over the domain of the first membership degree known as the secondary membership degree. This is formally expressed as:
\[
\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad , J_x \subseteq [0,1]
\] (2.23)

where \( \tilde{A} \) is the GT2 FS and \( \mu_{\tilde{A}}(x, u) \) is the membership function of \( \tilde{A} \). \( J_x \) is the primary membership of the input domain \( x \in X \) and \( u \in J_x \) [6]. As before, the \( \int \) symbol denotes all possible values over the specified domain [33].

There are several methods of representing the MF of a GT2 FS, here, the vertical-slice representation will be discussed, as it is the most widely used. For a specific value \( x' \) in the input domain, a vertical slice can be defined and its MF can be defined as follows:

\[
\mu_{\tilde{A}}(x', u) = \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) / u \quad , \forall u \in J_{x'} \subseteq [0,1]
\] (2.24)

where \( f_{x'}(u) \in [0,1] \) is the secondary MF at the input value \( x' \). Thus, the vertical-slice representation can be used to describe a GT2 MF as a set of slices over the input domain \( x \in X \):

\[
\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x) / x = \int_{x \in X} \left( \int_{u \in J_{x'}} f_{x'}(u) / u \right) / x \quad J_{x'}^u \subseteq [0,1]
\] (2.25)

The amount of uncertainty that is captured by a GT2 FS can be assessed using the concept of Footprint of Uncertainty (FOU) can be used. The FOU is defined as the bounded region created by taking the union of all primary MF of the GT2 FS:

\[
FOU(\tilde{A}) = \bigcup_{x \in X} J_x
\] (2.26)
2.2.2 Representations of GT2 FSs

As mentioned, the above described representation and type-reduction of GT2 FS is extremely computationally expensive [39]. Therefore, two other popular representations of GT2 FS have been proposed in literature, which are the z-slices representation [40], and the $\alpha$-planes representation [41], [42]. While both methods rely on similar slicing of the secondary MF, the $\alpha$-planes representation is used in this dissertation and therefore will be discussed here. Given a GT2 FS $\tilde{A}$ an $\alpha$-plane $\tilde{A}_\alpha$ can be defined as the union of all primary MFs of $\tilde{A}$ with secondary grade greater than or equal to $\alpha$ [41], [43]:

$$\tilde{A}_\alpha = \bigcap_{\forall x \in X} \bigcap_{\forall u \in J_x} \{(x,u) \mid f_x(u) \geq \alpha\}$$  \hspace{1cm} (2.27)

A $\alpha$-cut of the secondary MF of $\tilde{A}$, $\mu_{\tilde{A}}(x)$ is a vertical slice of the GT2 FS and is expressed using an interval as [41], [43]:

$$S_A(x \mid \alpha) = [s_L(x \mid \alpha), s_R(x \mid \alpha)]$$  \hspace{1cm} (2.28)

where $s_L$ and $s_R$ are the left and right boundaries of the secondary MF of $\tilde{A}$ that defines the $\alpha$-cut.

Thus a $\alpha$-plane is a collection of $\alpha$-cuts of all secondary MF for the complete domain of the input, which is represented as [41], [43]:

$$\tilde{A}_\alpha = \bigcap_{\forall x \in X} S_A(x \mid \alpha)$$  \hspace{1cm} (2.29)

which is equivalent to
\[ \tilde{A}_\alpha = \int_{x \in X} \left( \int_{u \in [s_x(s) u], s_x(s) u} u \right) / x \]  

(2.30)

Further, the \( \alpha \) -plane can be raised in the secondary domain of the original GT2 FS \( \tilde{A} \), to create a \( \alpha \) -level that is represented as [43]:

\[ R_{\tilde{A}_\alpha}(x,u) = \alpha / \tilde{A}_\alpha, \quad \forall x \in X, \forall u \in J_x \]  

(2.31)

Thus, a GT2 FS can now be represented as a collection of \( \alpha \) -levels of the original GT2 FS \( \tilde{A} \) as [42]:

\[ \tilde{A}' = \bigcup_{\alpha \in [0,1]} \alpha / \tilde{A}_\alpha \]  

(2.32)

where the \( \bigcup \) operator denotes the union of FS. Thus, the more \( \alpha \) -levels used, the more accurate the representation becomes.

\subsection*{2.2.3 Set Theoretic Operations on GT2 FSs}

As with T1 FSs, set theoretic operations that are analogous to classical Boolean set theory exist for GT2 FS. As before, only the operations that are relevant to this dissertation will be discussed here.

Given the GT2 FS \( \tilde{A} \) described in (2.25), and another GT2FS \( \tilde{B} \) described as:

\[ \tilde{B} = \int_{x \in X} \mu_{\tilde{B}}(x) / x = \int_{x \in X} \left( \int_{w \in J_x} g_x(w) / w \right) / x \quad J_x^\mu \subseteq [0,1] \]  

(2.33)

where the intersection operation can be defined as follows [6]:

20
Given the two GT2 FSs described above, the union of two GT2 FSs can be defined as:

$$\tilde{A} \cup \tilde{B} = \bigcup_{u \in J^*_1, w \in J^*_2} \left( \bigcup_{u \in J^*_1, w \in J^*_2} f_i(u) \cap g_i(w) \right) \forall x \in X$$  \hspace{1cm} (2.35)

where $\bigcup$ is the t-conorm or s-norm operation as described in 2.1.2. The difference in union function as compared to the intersection is that for union, the t-conorm is calculated for the primary MF of the two GT2 FSs. The complement of a GT2 FS is calculated as:

$$\overline{A} = \bigcup_{u \in J^*_1} f_i(u) \cap (1-u) \forall x \in X$$  \hspace{1cm} (2.36)
2.2.4 General Type-2 Fuzzy Logic Systems (GT2 FLSs)

A General Type-2 Fuzzy Logic System (GT2 FLS) is a fuzzy logic system with the same structure as a T1 FLS, but utilizes GT2 FSs for input or output representation. Figure 3 shows the basic structure of a GT2 FLS.

The fuzzification stage produces MF which are the secondary MF of the GT2 FS, as compared to membership degrees in T1 FLS. The rule structure is the same as in T1 FLS which is shown in Equation (2.19). The inference process utilizes the same logic expressed in Section 2.1.3 Equations (2.20) and (2.21). The difference here is the output of the inference is again a GT2 FS. The primary difference between the GT2 FLS and T1 FLS is producing the final output. In the GT2 FLS, an additional step is required to reduce the generated GT2 FS into a T1 FS before defuzzification. This step is called the type-reduction of the GT2 FS.

The basic method for type-reduction is the exhaustive method which is highly computationally expensive [6]. While, there are several algorithms proposed for type-reduction [6], [8], [10], these will not be discussed as they are out of the scope of this dissertation. The exhaustive method discretizes the input domain $X$ into $N$ samples. The primary MF of the GT2 FS at these discretized points are denoted as $J_{x_1},...,J_{x_y}$. Each of these primary MF $J_{x_i}$ are then further discretized in to $M_i$ points. Using this discretization embedded FS can be generated which has a single element from each $J_{x_1},...,J_{x_y}$, namely $\theta_1,\ldots,\theta_N$. Thus, from this discretization $\prod_{i=1}^{y} M_i$ number of embedded FS are generated. Processing this large number of embedded FS causes the high computational complexity of this method. The centroid of an embedded FS can be calculated using [6]:

$$\text{centroid} = \frac{\sum_{i=1}^{N} \theta_i}{N}$$
\[ a(\theta) = \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i} \]  

This centroid is calculated for each embedded FS of the initial GT2 FS. Furthermore, for each of the embedded FS, the t-morn of the secondary membership grade is calculated at the discretization intervals [6]:

\[ b(\theta) = f_{x_1}(\theta_1) \prod ... \prod f_{x_N}(\theta_N) \]  

Once these measures are calculated for all embedded FS, the values are paired according to the corresponding embedded FS. Finally, the maximum \( b(\theta) \) for a given unique input domain value \( a(\theta) \) is selected to generate the type-reduced FS. This can be written more formally as [6]:

\[ C_\tilde{A} = \int ... \int_{\theta_1 \in J_1, \theta_N \in J_N} b(\theta) / a(\theta) \]  

where \( C_\tilde{A} \) is the type-reduced T1 FS of the GT2 FS \( \tilde{A} \).

After the type-reduction is performed, defuzzification of the resulting T1 FS yields a scalar output value of the GT2 FLS. For defuzzification the methods described in section 2.1.3 can be used. The T1 FS \( C_\tilde{A} \) that results from the type-reduction can be used as an uncertainty measure which describes how uncertain we are about a given output value [6].
2.3 Interval Type-2 Fuzzy Logic (IT2 FL)

Because of the high computational complexity of the GT2 FSs, and GT2 FLSs, Interval Type-2 Fuzzy Logic (IT2 FL), was introduced and subsequently became popular among researchers [32], [44], [45]. IT2 FL is a special case of GT2 FL, where the secondary membership grade of the FS are represented as intervals rather than FS themselves. Thus, IT2 FL offers computationally efficient algorithms compared to GT2 FL, while maintaining some of its uncertainty modeling capabilities [6], [10], [19].

2.3.1 Interval Type-2 Fuzzy Sets (IT2 FSs)

As mentioned, a IT2 FS is a constrained GT2 FS where the secondary membership grade is always 1:

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

(2.40)

where $J_x$ is the primary membership of $\tilde{A}$ at input value $x$. This representation is analogous to the GT2 FS representation shown in (2.23). Thus, all secondary membership values of an IT2 FS are equal to 1 [6]. As with GT2 FSs, IT2 FSs can also be represented using the vertical slice representation. Similar to GT2 FSs this representation is defined for a specific value of the input $x'$:

$$\mu_{\tilde{A}}(x',u) = \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} 1/u \quad \forall u \in J_{x'} \subseteq [0,1]$$

(2.41)

This representation is analogous to the GT2 FS representation given in Equation (2.24). The Footprint of Uncertainty (FOU) can be defined for IT2 FS using Equation (2.26) as well.
Furthermore, two T1 MF are defined as upper MF $\mu_A$ and lower MF $\mu_A$, such that the upper MF is equal to the upper bound of the FOU and lower MF is equal to the lower bound of the FOU:

$$\mu_A(x) = FOU(\tilde{A}), \forall x \in X$$  \hspace{1cm} (2.42)

$$\mu_A(x) = FOU(\tilde{A}), \forall x \in X$$  \hspace{1cm} (2.43)

Thus, the primary MF $J_z$ of an IT2 FS $\tilde{A}$ can be fully defined using the upper MF and lower MF [6], [32]:

$$J_z = [\mu_A(x), \mu_A(x)], \forall x \in X$$  \hspace{1cm} (2.44)

Thus, an IT2 FS can be represented as [6], [32]:

$$\tilde{A} = \int_{x \in X} 1/[\mu_A(x), \mu_A(x)]$$  \hspace{1cm} (2.45)

The possibility of representing a IT2 FS using two T1 FSs enables the better computation time compared to GT2 FSs.

### 2.3.2 Set Theoretic Operations on IT2 FSs

As before, the basic set theoretic operations of IT2 FSs are analogous to Boolean set theoretic operations. Using the IT2 FS notation in (2.45) each of these operations can be calculated by applying a given operation to upper and lower MF separately. Given two IT2 FSs $\tilde{A}$ and $\tilde{B}$ (represented using the notation in (2.45)), the intersection can be defined as [6]:

...
\[
\tilde{A} \cap \tilde{B} = \frac{1}{\mu_{\tilde{A}}(x) \cap \mu_{\tilde{B}}(x), \mu_{\tilde{A}}(x) \cap \mu_{\tilde{B}}(x)} \quad \forall x \in X \tag{2.46}
\]

Similarly, the union can be defined as:

\[
\tilde{A} \cup \tilde{B} = \frac{1}{\mu_{\tilde{A}}(x) \cup \mu_{\tilde{B}}(x), \mu_{\tilde{A}}(x) \cup \mu_{\tilde{B}}(x)} \quad \forall x \in X \tag{2.47}
\]

The complement is defined as:

\[
\overline{\tilde{A}} = \frac{1}{1 - \mu_{\tilde{A}}(x), 1 - \mu_{\tilde{A}}(x)} \quad \forall x \in X \tag{2.48}
\]

Thus, by applying the operations to two T1 FSs computationally efficient calculations can be made by using IT2 FSs compared to GT2 FSs.

### 2.3.3 Interval Type-2 Fuzzy Logic Systems (IT2 FLSs)

The IT2 FLS takes the same form as a GT2 FLS and Figure 3 applies to IT2 FLS as well. The only difference is the result of type-reducing a IT2 FS is an interval rather than a T1 FS as is the case in GT2 FLS. This interval can be expressed as \( C_{\tilde{B}} = [c_i, c_r] \), where \( c_i \) and \( c_r \) are the left and right boundaries respectively.

The type-reduction of IT2 FS takes the same form as GT2 FS type reduction. However, several algorithms have been proposed that takes advantage of the IT2 FS representation described in (2.38) to compute the type-reduction more efficiently [39], [46]-[53]. While each method approaches the problem differently, each method has its own strengths and weaknesses. This section will briefly discuss one of the most widely used methods with is the Karnik-Mendel (KM) method [39]. In the KM method, the two boundaries \( c_i \) and \( c_r \) are
calculated using two switch points in the upper and lower MF. The boundaries can be defined as:

\[
c_i = \frac{\sum_{i=1}^{L} x_i \mu_A(x_i) + \sum_{i=L+1}^{N} x_i \mu_A^{-1}(x_i)}{\sum_{i=1}^{L} \mu_A^{-1}(x_i) + \sum_{i=L+1}^{N} \mu_A(x_i)} \quad \text{(2.49)}
\]

\[
c_r = \frac{\sum_{i=1}^{R} x_i \mu_A^{-1}(x_i) + \sum_{i=R+1}^{N} x_i \mu_A(x_i)}{\sum_{i=1}^{R} \mu_A(x_i) + \sum_{i=R+1}^{N} \mu_A^{-1}(x_i)} \quad \text{(2.50)}
\]

where \( N \) is the number of discretization in the input domain. \( L \) and \( R \) are the switching points that will be calculated using an iterative method described below \([39], [46], [47] \). To find the \( L \) and \( R \) switching points, first an initial point is calculated using:

\[
c' = \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i} \quad \text{(2.51)}
\]

where \( \theta_i = \frac{1}{2} (\mu_A(x_i) + \mu_A^{-1}(x_i)) \), \( i = 1, 2, \ldots, N \), and \( c' \) is the candidate centroid boundary. Second, the candidate switching \( k \) point can be found such that \( 1 \leq k \leq N-1 \) and \( x_k \leq c' \leq x_{k+1} \). Once, this is done, the \( \theta \) values are updated as:

\[
\theta_i = \begin{cases} 
\mu_A^{-1}(x_i) & , i \leq k \\
\mu_A(x_i) & , i \geq k + 1
\end{cases}
\quad \text{(2.52)}
\]

Using the updated \( \theta \) values, a new candidate centroid boundary \( c'' \) is calculated using Equation (2.44). If the new candidate centroid boundary is equal to the previous candidate
centroid boundary, i.e. if \( c' = c'' \), then the algorithm is halted and the left centroid \( c_l \) boundary and the left switching point \( L \) are set as \( c_i = c' \) and \( L = k \). If \( c' \neq c'' \), then the candidate boundary is set to \( c' = c'' \) and a new candidate switching point is found. This is iterated until the condition \( c' = c'' \) is satisfied. Calculating the right centroid boundary \( c_r \) is identical to above except Equation (2.45) is updated as:

\[
\theta_i = \begin{cases} 
\frac{\mu_{\mu}^A(x_i)}{i \leq k}, \\
\frac{\bar{\mu}_{\mu}^A(x_i)}{i \geq k + 1}
\end{cases}
\] (2.53)

The final crisp output of the IT2 FLS is calculated by calculating the average of the type-reduced centroid. As with GT2 FLS, the output centroid is a direct measure of uncertainty of the output value produced by the IT2 FLS [6].

### 2.4 Conclusions

This chapter introduced T1 and T2 FL, as well as FSs and fuzzy arithmetic that is relevant to this dissertation. T1 and T2 FLS, which are the most widely used applications of FL was also discussed. Furthermore, how understandability and uncertainty modeling is related to each type of FS was also discussed.

Furthermore, the understandability of FL stems from the capability of modeling real valued data using human understandable prototypes which are MF. The understandability further increases by means of utilizing and computing based on linguistic consequential rules. The uncertainty modeling of FL is increased by the use of GT2 FS, however, as mentioned,
the computational complexity of GT2 FLS is extremely high, and therefore, the more computationally efficient IT2 FLS is widely used. However, the lower uncertainty modeling capability of IT2 FS when compared to GT2 FS has to be noted.

It has to be noted that as fuzzy logic has been a heavily researched area for over 5 decades, various different algorithms and representations exist for better performance and computational complexity. The methods and algorithms described in this chapter are only the ones that are relevant to the work presented in this dissertation.
CHAPTER 3

GENERATING FUZZY SETS FOR UNDERSTANDABILITY

The primary advantage of using FSs and FLSs is the utilization human understandable linguistic terms that are capable of capturing uncertainty and vagueness in everyday language [54]. The ability to handle such linguistic terms make FLS attractive as they are highly interpretable, transparent, and understandable [12], [55]-[57]. The understandability of a fuzzy rule can be defined as the extent to which a fuzzy rule can be read and understood constrained to whether the correct knowledge is understood by the reader [14], [21]-[23]. Thus, the linguistic terms used in the fuzzy rule must be consistent to data, expert opinions and human intuition [11], [14], [23].

To capture the uncertainty and vagueness inherent to linguistic terms, Fuzzy Membership Functions (MF) are used. Therefore, MF are essential for improving the understandability of FLSs. Optimizing FLSs for improved accuracy in terms of classification or control can reduce the understandability of the generated fuzzy MFs, and subsequently reduce the understandability of the whole FLS. Expert knowledge can be used to derive MFs, but it has been shown that this might not be optimal, and acquiring expert knowledge is not time consuming and not trivial.

Therefore, a data driven method using statistical methods to generate MFs that describe the data while maintaining the understandability, is presented in this Chapter. The presented method utilizes statistical techniques to calculate key points of MF such as centers, spread,
overlap, slope etc. Student’s t-test is utilized to identify initial prototypes for MFs. Two different understandability metrics are introduced that measures understandability of MFs. Further, the presented MF generation method utilizes these and several other understandability metrics to generate MFs and thus further increasing the understandability of the MFs. The presented method was tested on several benchmark datasets with known distributions as well as a real-world dataset. The generated MFs were shown to describe the data while maintaining high levels of understandability.

3.1 Understandability of Fuzzy Systems

While using human understandable linguistic terms increase the understandability and interpretability of FLSs, the understandability of a FLS is heavily dependent on the understandability of its linguistic terms [14], [22]. The linguistic terms are modeled in the data domain via Fuzzy Membership Functions (MFs). Therefore, determination of the linguistic terms, hence the generation of MFs plays a significant role fuzzy system design [14]. The MFs used in the FLS should be capable of conveying the knowledge contained in the original data in a relevant and understandable manner [34], [23]. For example, in a dataset related to building heating and cooling, the MF “hot” should be in the range of around 75°F and higher, furthermore, the MF should not span the complete range of the data. Similarly, for a control system for a high-temperature furnace the MF “hot” will take much higher values. Thus, domain dependent data driven MF generation is crucial for obtaining relevant MF [11].

Furthermore, the design, meaning size, shape and position of MFs significantly affect the performance of the FLS [58], [59]. Therefore, many recent work optimize FLS by changing
MF parameters by simply focusing on the accuracy of the output without considering the understandability of the system [55], [60]-[63]. Such numerical optimization results in highly accurate FLS with good performance characteristics; however, they pay little attention to the semantical properties of the generated MFs and linguistic terms, degrading the understandability of the resulting system [15], [35], [55].

For data mining applications such as Linguistic Summarization (LS) [15], [64], [65] or descriptive rule generation [66], where interpretability and understandability of data is the goal [65], generating semantically correct MFs is important. One of the major reasons for the overlooking of understandability when designing FLSs is because understandability can be subjective and domain specific. Therefore, formalizing and measuring understandability or interpretability is a difficult task [22] [23].

### 3.2 Generating Fuzzy Membership Functions

Typical MF generation techniques include data histogram based methods, heuristic optimization based methods, probability based methods, clustering based methods, neural networks based methods, etc. [67]. However, all these techniques focus on performance improvement of the final FLS and very little work focuses on the understandability of generated MF [14], [35], [56], [61]. Furthermore, it has been shown that there is no one optimal way of generating MFs and the optimality depends on the application, specific data and requirements [67].

Membership functions derived from expert knowledge are capable of solving understandability issues [59]. However, expert knowledge acquisition can be a difficult and
time consuming task. Experts on the required domain may not always be available, and even when they are available their opinions can be incomplete, varying, or overly precise [56], [35]. The lack of experience in Fuzzy Logic, on the behalf of the domain expert is also a hindrance. Furthermore, due to the large number of dimensions, gathering expert knowledge for highly multi-dimensional problems is difficult.

Many applications also consider pre-defined fuzzy MF [14]. However, this is also sub-optimal as it assumes the data will be distributed in a certain manner and therefore cannot be effectively used to handle specificity of real-world problems [14]. Therefore, the most attractive method of deriving MF is data driven. Furthermore, data driven methods have been shown to accommodate adaptation and self-tuning [56].

3.3 Generating Fuzzy Membership Functions for Understandability

The MF generation methodology presented in this chapter is a data driven technique that maximizes several understandability metrics to generate the desired MFs for a given dataset. This section first introduces the understandability metrics and the reasoning behind each metric. Further, two new understandability metrics are also introduced. Second, related work that generates MF for increased understandability is detailed. Third, the presented MF generation method is detailed.
3.3.1 Understandability Metrics

Fuzzy Membership Functions (MFs) capture the uncertainty and vagueness of everyday linguistic terms [54]. Therefore, finding optimal partition of the input space, the shape of MFs, the coverage of MFs, and the linguistic term associated with the MFs are significant factors affecting the understandability of FLS [14], [23], [37], [60], [68]. Since understandability or interpretability depends on multiple factors, and understandability is extremely subjective and domain dependent, metrics for measuring true understandability are difficult to define [22], [23]. Nevertheless, many metrics have been proposed in literature that measure understandability or interpretability of MFs [22], [23], [35], [60], [69].

Figure 4 demonstrates important parameters of MFs defined for the input dimension $X$, using trapezoidal MFs. For a given input value $x$ the membership degree for each MF $\mu_i(x)$,
where $i$ is the MF, can be calculated. The core or the center of the MF is defined as the set of values whose membership degree is 1: $\forall x \in \mu_i(x) = 1$. Similarly, the footprint or the support of the MF is where the membership degree is greater than 0: $\forall x \in \mu_i(x) > 0$. The core and the support of a MF can be expressed using the following values:

$$C_{L,i} = \min(x: \mu_i(x) = 1) \ \forall \ x \in X$$ (3.1)
$$S_{L,i} = \max(x: \mu_i(x) = 0) \ \forall \ x \in X$$ (3.2)
$$C_{R,i} = \max(x: \mu_i(x) = 1) \ \forall \ x \in X$$ (3.3)
$$S_{R,i} = \min(x: \mu_i(x) = 0) \ \forall \ x \in X$$ (3.4)

Equations (3.1) and (3.3) describe the left and right boundaries of the core, respectively. Equations (3.2) and (3.4) describe the left and right boundaries of the footprint of the MF respectively. Thus, the core of the MF $i$ can be expressed as:

$$\text{Core}_i = C_{L,i} \leq x \leq C_{R,i} \ \forall \ x \in X$$ (3.5)

Similarly, the support of the MF $i$ can be expressed as:

$$\text{Support}_i = S_{L,i} < x < S_{R,i} \ \forall \ x \in X$$ (3.6)

A Normal MF is defined as a MF that has a core, i.e. the membership degree is 1 for at least one point in the input dataset, and is considered to be a property for increased understandability [22], [34], [35]. For a given MF $i$, Normality can be expressed as:

$$\text{norm}_i = \max(\mu_i(x)), \forall x \in X$$ (3.7)

where $\text{norm}_i$ is the normality for $i^{th}$ MF. Thus, if a given MF has a core, then normality will be 1.
It is universally agreed that MFs should be monotonic and convex for increased understandability [34], [22], [54], [62], [63]. In [68], this property is referred to as *Unimodality*, which can be expressed as:

\[ \forall x_1, x_2, x_3 \in X : x_1 < x_2 < x_3 \rightarrow \mu_i(x_2) \geq \min\{\mu_i(x_1), \mu_i(x_3)\} \]  

Dataset *Coverage* is also considered by many as a simple yet important metric for understandability [22], [35], [69]. This metric states that the range of the input dataset should be covered by at least one MF to a certain degree. Coverage can be expressed as:

\[ \forall x \in X, \mu_i(x) > \beta \]  

where \( 0 < \beta \leq 1 \) and can be preset by the user, and \( \forall i \in p \) where \( p \) is the set of MF for input dimension \( X \).

Relatively moderate number of MFs for each dimension is also important for understandability [22], [35], [55]. This characteristic is simplified by authors in [55] as Partition Granularity by:

\[ part = \frac{1}{p-1} \]  

where \( p \) is the number of MFs for a given dimension, and it is assumed that \( p \geq 2 \).

For increased understandability, generated MFs should be sufficiently distinct from each other, with limited overlap [22], [35], [69]. This is typically achieved by a threshold value for intersection points:

\[ \forall i, j \in p : \mu_i(x) = \mu_j(x) \iff \mu_i(x) < \alpha \]
where $0 < \alpha < 1$ and can be preset by the user, and $p$ is the number of MFs.

The measure Complementarity [22] is also closely related to the above measure. Complementarity is a property where for a given input value, the sum of all membership degrees is close to 1, and can be expressed as:

$$\forall x \in X : (1 - \delta) < \sum_{i=1}^{p} \mu_{i}(x) < (1 + \delta)$$

(3.12)

where $\delta \approx 0$ and can be preset by the user.

Another measure used to identify the distinctness of MFs is the property of Separation. This measure is different from distinct measure above as this property states that the cores of adjacent MFs must be separated by at least $\eta$, where $\eta$ is a value in the input domain set by the user. This can be expressed as:

$$\forall i \in p; \left| C_{R,i} - C_{L,j} \right| \geq \eta; \forall j \neq i \in p$$

(3.13)

Symmetry of generated MF is also considered as a relative measure of understandability as it reflects of universal duality and natural relativity of terms [22], [70]. In this dissertation, the measurement, Relative Dissymmetry is presented to measure symmetry in MFs. First, the dissymmetry measure is formalized as:

$$diss_{i} = \left| \sum_{x=0}^{s_{R,i}} \mu_{i}(x) - \sum_{x=0}^{s_{L,i}} \mu_{i}(x) \right|$$

(3.14)

where $diss_{i}$ is the dissymmetry measure for $i^{th}$ MF, and the dissymmetry measure can be normalized for a given MF as:
\[
\overline{diss_i} = \frac{diss_i}{\max \left( \sum_{x=S_{L,i}} \mu_i(x), \sum_{x=C_{R,i}} \mu_i(x) \right)}
\]

(3.15)

where \( \overline{diss_i} \) is the Relative Dissymmetry of the MF \( i \). The dissymmetry measure cannot be calculated for shoulder MFs that describe extremes (minimum and maximum) of data.

A threshold can also be set for the membership degree of a MF inside the core of another MF \([22],[60]\). This property can be formalized as:

\[
\forall x \in [C_{L,j}, C_{R,j}], \mu_j(x) < \gamma \forall j \neq i \in p
\]

(3.16)

where \( 0 \leq \gamma \leq 1 \) and can be preset by the user. Typically \( \gamma \) is set to zero \([60]\), this means that the membership degree of other MFs is zero within the core of a given MF.

Finally, in this dissertation, a measurement is introduced that measures the compliance of a MF to the data explained by it. This measure is called Compliance and can be measured by utilizing a normalized histogram \([71]\) containing \( n \) bins. The normalized degree of compliance for \( i^{th} \) MF \( \overline{comp_i} \) can be formalized as:

\[
\overline{comp_i} = 1 - \frac{uncomp}{\sum_{x=S_{L,i}} \hat{h}(x)}
\]

(3.17)

where \( \hat{h}(x) \) is the maximum possible value of the histogram for the input value \( x \), which, in the case of the normalized histogram is 1. In addition,

\[
uncomp = \sum_{x=S_{L,i}} |h(x) - \mu_i(x) |
\]

(3.18)
where, $h(x)$ is the height of the normalized histogram for value $x$. The normalized degree of compliance is 1 when the MF fully complies with the data and decreases if the MF does not comply with the data.

### 3.3.2 Related Work

Several authors have proposed methods for generating understandable MFs in recent years [22]. Some have used several of the above mentioned understandability (interpretability) measures. Typically the previously proposed methods used can be separated into three main categories: cluster based methods, evolutionary methods, and combined methods with preset MFs.

Most cluster based method utilize a form of Fuzzy C-Means (FCM) to generate the MF. A constrained FCM based method is used in [54] generating more understandable MFs. Similarly, a modified FCM based method is used in [68]. The authors utilize clustering to generate MFs and then combine similar MFs for increased understandability in [56] and [72]. Clustering and cluster distances are used to derive understandable MFs in [69]. However, clustering based methods has several disadvantages as well as advantages [54].

Evolutionary algorithm based techniques have been proposed that optimizes MF based on one or more understandability metrics. In [14] the authors use hedge algebra based semantics to assign linguistic terms to information granules and utilize simulated annealing to optimize MFs. Symmetrical MFs are generated using evolutionary algorithms in [70]. In [60] and [35] evolutionary algorithm based approaches are proposed that make use of understandability metrics.
The authors use preset MFs, and tune these using lateral movements in [73]. In [62] and [63] the authors propose an algorithm that utilizes pre-shaped MFs along with FCM clustering to generate transparent MFs. Similar method that also utilizes evolutionary algorithms is proposed in [61] and [74]. Preset MFs require the use of experts and may be sub-optimal as mentioned before. The primary drawback of clustering and evolutionary algorithm based methods is the increased computational complexity.

In contrast, the method presented in this dissertation utilizes a deterministic, statistical approach to identify the optimal parameters for MFs. Furthermore, the presented method utilizes several understandability metrics to derive and fine-tune these parameters. This ensures that the understandability of generated MFs is maintained, while fully describing the data distribution.

### 3.3.3 Presented Data Driven Method for Generating Membership Functions

The presented data driven, statistics based MF generation method is a 5 step process. Each step can be summarized as follows:

- **Step1**: Generate initial prototypes,
- **Step2**: Refine generated prototypes,
- **Step3**: Generate initial MFs,
- **Step4**: Remove unwanted MFs, and
- **Step5**: Refine generated MFs.
Each step is focused on increasing the understandability of the MFs while maintaining the ability to describe the data properly. Prior to the MF generation process the dataset is normalized between 0 and 1. Detailed descriptions of each step are given below.

**Step 1**: In the first step initial prototypes for MFs are generated. The first prototypes are the sample mean \( \bar{X}_{\text{data}} \), minimum, \( \min_{\text{data}} \) and maximum, \( \max_{\text{data}} \) values of the dataset. The Student’s t-test for unequal sample size and unequal variance [75] was used to identify portions of the data that are significantly different from the mean, to generate secondary prototypes. The Student’s t-test for two mean values \( \bar{x}_1 \) and \( \bar{x}_2 \) can be expressed as:

\[
t(x_1, x_2) = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}
\]

(3.19)

where, \( s \) and \( n \) are standard deviation and sample size of each sample, respectively. If the \( t \) value is greater than the critical \( t \) value then the null hypothesis is rejected, meaning the two sample means are significantly different from each other. Using the Student’s t-test two prototypes to the left and two prototypes to the right of the sample mean are generated. These are formalized as:

\[
PB_L = \max(x): \begin{array}{l}
0 < x < \bar{X}_{\text{data}}, \\
t(x, \bar{X}_{\text{data}}) > t_{\text{Critical}}
\end{array}
\]

(3.20)

\[
PM_L = x_{0, PB_L}
\]

(3.21)

Similarly,
\[ PB_r = \min(x) : \]
\[
\bar{X}_{data} < x < 1,
\]
\[
t(x_{t,1}, \bar{X}_{data}) > t_{critical}
\]
\[
P_{M_r} = \bar{X}_{PB, l}
\]

(3.22)

where \( PB_L \) and \( PM_L \) are two secondary prototypes to the left of the sample mean and \( PB_R \) and \( PM_R \) are two prototypes to the right of the sample mean. These prototypes signify portions of the data that are significantly different from the initial prototype (sample mean \( \bar{X}_{data} \)).

This process is iterated to the left of \( PB_L \), and to the right of \( PB_R \) using \( PM_L \) and \( PM_R \) as the initial prototypes, until the newly generated prototypes surpasses the minimum and maximum values.

**Step 2:** In this step the generated initial prototypes are refined. First similar prototypes are combined. Prototypes within \( \varepsilon \) range of each other are averaged:

\[
\forall i \neq j \in k, P_n = \frac{P_i + P_j}{2} \Rightarrow |P_i - P_j| < \varepsilon
\]

(3.24)

where \( k \) is the set of generated prototypes, \( P_i \) and \( P_j \) are prototypes \( P_n \) is the new prototype, \( \varepsilon \) is a preset constant, and \( | \cdot | \) denotes the absolute value. Once this is done, \( P_i \) and \( P_j \) are removed from \( k \) and \( P_n \) is added to \( k \).

Secondly, prototypes within \( \varepsilon \) of the minimum and maximum values are removed, retaining only the minimum and maximum values. This is done because it is generally accepted that data extremes must be prototypes of some MFs [69].

\[
P_i = \min_{data} \Rightarrow |P_i - \min_{data}| < \varepsilon
\]

(3.25)
\[ P_i = \max_{data} \Rightarrow |P_i - \max_{data}| < \varepsilon \]  

(3.26)

Thirdly, the remaining prototypes are grouped to satisfy the separation property described in equation (3.13).

\[
\forall i \in k, |P_i - P_{i+1}| < \eta \Rightarrow G_m = \{P_i, P_{i+1}\} 
\]  

(3.27)

\[
\forall i \in k, |P_i - P_{i+1}| \geq \eta \Rightarrow G_m = \{P_i\}, G_{m+1} = \{P_{i+1}\} 
\]  

(3.28)

where, \(G\) is a set of prototypes. The grouping is performed in ascending order of the distances between prototypes ensuring the closest prototypes are grouped together. The grouping process ensures, data that is separable but leads to higher granularity is combined and represented by one MF at a later stage. Furthermore, during grouping if a group contains more than three prototypes, prototype that is closest to the left or right boundary of the group is deleted. This grouping process is performed until all prototypes satisfy the conditions given in Equations (3.27) and (3.28). However, it is ensured that the groups containing the minimum and maximum prototypes of the data are not grouped together [69]. This implies that this step will be terminated if there are only two groups left.

**Step3**: Once the prototypes are refined, the initial MF are generated. For each of the generated groups \(G\) a MF is generated. The cores of the MFs are defined using the minimum and maximum prototypes of a group \(G\):

\[
C_{L,i} = \min(P_j) \forall P_j \in G; i = 1,2,...,M 
\]  

(3.29)

\[
C_{R,i} = \max(P_j) \forall P_j \in G; i = 1,2,...,M 
\]  

(3.30)

where, \(i\) is the generated MF and \(M\) is the set of generated groups in Step2. The support of the \(i^{th}\) MF is defined as:
\[ S_{L,i} = \max(P_j) \forall P_j \in G_k : P_j < C_{L,i} ; k = 1,2,\ldots,M \]  
(3.31)  
\[ S_{R,i} = \min(P_j) \forall P_j \in G_k : P_j > C_{R,i} ; k = 1,2,\ldots,M \]  
(3.32)  

Thus, the left support of the \( i^{th} \) MF \( S_{L,i} \) is the prototype to the immediate left of the left core \( C_{L,i} \), and similarly the right support \( S_{R,i} \) is the prototype to the immediate right of the right core \( C_{R,i} \).

**Step4:** In this step the some of the generated MF are removed or combined to decrease granularity and increase understanding. An average understandability is defined for the fuzzy system which is used to identify MFs that will be removed or combined. The average understandability, \( AU \) is defined in terms of normalized dissymmetry described by Equation (3.15) \( \overline{diss} \) and normalized degree of compliance described by Equation (3.17) \( \overline{comp} \). The normalized dissymmetry and the normalized degree of compliance can now be used as the averaged understandability because all the remaining metrics have already fulfilled by the prototypes that are remaining. The averaged understandability \( (AU) \) can be defined as:

\[
AU = \left( \frac{\sum_{i=1}^{M} \overline{diss}_i}{M} + \frac{\sum_{i=1}^{M} \overline{comp}_i}{M} \right) \div 2
\]  
(3.33)  

where \( M \) is the number of MFs in the FLS, and \( AU \) is the averaged understandability.

When a MF is deleted, the prototypes that were used to generate the core of that MF are also deleted. For certain situations detailed below, MFs are combined. This is done by
creating a new MF by defining the core as the leftmost and rightmost prototypes of the MFs that are combined:

\[
C_{L,n} = \min(P_k) \forall P_k \in G_i, G_j
\]

\[
C_{R,n} = \max(P_k) \forall P_k \in G_i, G_j
\]

where \(i\) and \(j\) are the MFs that are being combined and \(n\) is the new MF that is generated.

The MFs generated using the minimum and maximum prototypes of data are not deleted [69]. The remaining MFs are deleted or removed as follows: remove MFs that increases \(AU\) while the number of MF is greater than \(\omega\). If the number of MFs is still greater than \(\Omega\) then identify MF that reduces \(AU\) the least, and combine it with the closest MF using Equations (3.34) and (3.35), until the number of MF is less than or equal to \(\Omega\). The constants \(\omega\) and \(\Omega\) are user selected values such that \(2 \leq \omega \leq \Omega\). This ensures that the generated fuzzy system contains at least \(\omega\) MFs and less than or equal to \(\Omega\) MFs, ensuring the granularity constraints discussed earlier.

**Step5**: Finally, the spread of the remaining MFs are adjusted to fulfill the coverage criterion expressed by Equation (3.9), the sufficient distinct criterion expressed by Equation (3.11), the complementarity criterion expressed by Equation (3.12) and the membership degree inside another core criterion expressed by Equation (3.16). This is achieved using the same method in Step3 using Equations (3.31) and (3.32).
3.3.4 Convergence of the Presented Method

The convergence of the presented method can be defined in terms of whether the method terminates and generates MFs, and whether the generated MFs conform to the understandability metrics described. It has to be noted that the final outcome of the method is highly dependent on the preset values of the understandability metrics (shown in Table 1), and the closeness value $\varepsilon$ in Step 2. Nevertheless, the presented method should generate MFs in accordance to the preset values.

First the termination of the method is evaluated. It can be observed that each step of the presented method has finite iterations. The worst case scenario for Step 1 is that a prototype is generated for every data point in the dataset. Step 2 will terminate once there are no prototypes within $\eta$ range of each other, or when there is only two groups left. The remaining steps are bound by the number of groups generated by Step 2. As this is a finite number, Steps 3-5 are also finite. Thus, the termination condition of the presented method is guaranteed.

Second, the condition whether MFs are generated when the method terminates is evaluated. It is guaranteed that in Step 1 at least 2 prototypes will be generated as minimum and maximum bounds are in the set of initial prototypes. If in Step 2 all prototypes are removed or if all prototypes are grouped together no MFs will be generated. However, this situation is avoided as the minimum and maximum prototypes are not deleted and always will be grouped separately, ensuring at least two groups of prototypes at the end of Step 2. In Step 4, MFs are removed to increase the overall understandability. As $\omega$ and $\Omega$ have a lower bound of 2, after this step at least 2 MFs will remain. Thus, At least two MFs are guaranteed to be generated at the termination of the presented method.
Finally, the optimization of the understandability metrics is evaluated. The metrics normality and unimodality are guaranteed by the nature of the prototypes and the simple trapezoidal MFs utilized. Dataset coverage is guaranteed because the minimum and maximum bounds will always be prototypes and the initial MF generation in Step 3 and refinement in Step 5 guarantees all data points are covered. The partition granularity is guaranteed by the preset $\omega$ and $\Omega$ values. The metrics sufficient distinction, complementarity and property of separation are closely related together. These metric are optimized by the prototype refinement in Step 2 and final MF refinement if Step 5. Steps 3 and 5 guarantee that the property of a limited membership degree within a core of another MF is not violated. The measures relative dissymmetry and compliance are addressed and optimized in Step 4.

Thus, it can be concluded that the presented method converges to a solution and the solution is optimized for the given understandability metrics.

### 3.4 Experimental Results

The presented method was tested on several benchmark datasets with known distributions and a real world dataset. In order to verify the understandability of the generated MF, the compliance of each MF to the understandability metrics presented in Section 3.3.1 was used. This section first details the specifics of the experimental setup used, and then presents the experimental results.
3.4.1 Experimental Setup

In [12] Takagi and Sugeno stated that in order to claim the validity of a generated fuzzy system and the linguistic terms, the MFs must remain same in the presence of noise. Thus, in order to evaluate the validity of the presented system, different levels of noise were introduced to the dataset and MFs were generated. These MFs were then compared to the MFs generated without noise by using the following measure:

\[
\text{sim}_{q,r} = \frac{\sum_{x} \min(\mu_q(x), \mu_r(x))}{\sum_{x} \max(\mu_q(x), \mu_r(x))}
\] (3.36)

where, \(q\) and \(r\) are the MF that are being compared, and \(i\) is from the set of original MF and \(r\) is from the set of MF created with noisy data. The value \(\text{sim}_{q,r}\) is a measure of how similar MF \(q\) is to MF \(r\). The similarity value, sim is 1 when \(q\) and \(r\) are completely overlapped and 0 when there is no overlap. The similarity of the overall fuzzy system with all MFs was calculated using:

\[
\text{sys}_{ori,noise} = \frac{\sum_{r \neq q} \max(\text{sim}_{q,r}) \forall q \in p_{ori}}{p_{noise}}
\] (3.37)

where \(ori\) is set of MF generated using the original data and \(noise\) is the set of MF generated using the noisy data. \(p_{ori}\) is the number of MF in \(ori\) and \(p_{noise}\) is the number of MF in \(noise\).

As before, the value \(\text{sys}_{ori,noise}\) gives a similarity measure for the overall FLS generated with noisy data compared to the FLS generated with non-noisy data.
Noise from Signal-to-Noise Ratio (SNR) 20dB to SNR 10dB was introduced to the original data to generate the noisy data, where SNR is defined as:

$$SNR_{dB} = 10\log_{10} \left( \frac{A_{signal}}{A_{noise}} \right)$$  \hspace{1cm} (3.38)

where $A_{signal}$ and $A_{noise}$ are amplitude of the input and amplitude of the noise, respectively.

The preset values as tested for the understandability constraints presented in Section 3.3.1 are shown in Table 1. Using these preset values, four different benchmark datasets were tested. Each dataset was generated using a random number generator to follow a known distribution and contained 2000 data points. The tested known distributions were: uniform distribution, normal distribution, bivariate normal distribution, and right skewed distribution.

For further validation, the presented method was tested on a real-world dataset obtained form an office building containing temperature of 3 zones. These zones were selected for their varying distributions of data. The dataset contained temperature values collected every 30 minutes for a period of a month, thus containing nearly 1500 data points each.
Table 2. Similarity of the generated FLS at different noise levels

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Noise Level (SNR)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20dB</td>
</tr>
<tr>
<td>Benchmark Uniform</td>
<td>1</td>
</tr>
<tr>
<td>Benchmark Normal</td>
<td>0.99</td>
</tr>
<tr>
<td>Benchmark Bivariate</td>
<td>0.98</td>
</tr>
<tr>
<td>Benchmark Skewed</td>
<td>0.99</td>
</tr>
<tr>
<td>Real-world Zone1</td>
<td>0.9</td>
</tr>
<tr>
<td>Real-world Zone2</td>
<td>0.99</td>
</tr>
<tr>
<td>Real-world Zone3</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 5. Generated prototypes for the benchmark bivariate dataset at each step ((a)-(c)) and the final generated MFs (d)
Figure 6. Initial prototypes for (a) uniformly distributed data, (c) normally distributed data, and (e) right skewed data, along with generated MF for, (b) uniformly distributed data, (d) normally distributed data, and (f) right skewed data.
Figure 7. Initial prototypes for real world zone temperature data - (a), (c) and (e), along with generated MF - (b), (d) and, (f)
3.4.2 Experimental Results

For all cases tested, all understandability constraints given in Section 3.3.1 were met. This is because the presented method ensures each of the understandability constraints is met during the MF generation process. Thus, the generated MFs were at the highest level of measurable understandability in terms of understandability metrics.

The similarity of the generated fuzzy systems for different noise levels is shown in Table 2. Even for high noise levels, the generated fuzzy systems are close to the ones generated for the original dataset, thus confirming that the generated fuzzy systems are valid and describe the data consistently.

Figure 5 shows the generated prototypes for the benchmark bivariate dataset at each step. Step 4 is not shown because there are no major changes occurring at Step 5 to the prototypes. The initial prototype creation and grouping process can be observed in Figure 5 (a) and Figure 5 (b) respectively. Using the initial grouped prototypes initial MFs are generated in step 3 as shown in Figure 5 (c). Finally, one of the MFs initially generated is deleted and the supports of the remaining prototypes are adjusted in Steps 4 and 5 which is shown in Figure 5 (d).

The initial prototypes and the final generated MFs for each of the remaining benchmark datasets is shown in Figure 6. Similarly, the initial prototypes and the final MFs, for the real-world temperature dataset is shown in Figure 7.

Thus, it can be observed that the presented method generates intuitive and valid MFs for different data distributions while maintaining the overall understandability of the generated MFs.
3.5 Conclusions

This chapter presented a novel, data driven, statistical method for generating understandable Fuzzy Membership Functions (MFs) that describe data. The presented method uses classical statistical methods to identify initial MF prototypes. Two understandability measures were introduced. These, along with several other understandability metrics were used to generate and fine-tune MFs for increased understandability. The presented method was tested on several benchmark datasets with known distributions as well as real-world datasets. The presented method was shown to produce meaningful MFs that describe the data while maintaining high degree of understandability.

As future work, the presented method can be compared to other methods for understandable MF generation. The presented method can also be expanded further using more advanced statistical methods and will extended to accommodate the generation of Gaussian or Bezier MFs. Furthermore, the presented method can be extended to facilitate the generation of type-2 MFs. Hybrid of classical methods and the presented method can be used to improve accuracy in classification and control system FLS while maintaining high levels of understandability.
Descriptive and associative data summarization [65], [76] for understandability and interpretability that leads to increased state awareness of systems and data is gaining widespread use and benefits greatly from Fuzzy Logic [17], [65], [76]. This is known as Linguistic Summarization (LS) of data. Linguistic Summarization (LS) is used in data mining to extract trends and interdependencies present in large datasets. By using (LS), it is possible to obtain linguistic descriptions and rules that contain knowledge about the dataset which are also understandable to humans. Two types of linguistic summarization techniques can be used to summarize data: “Yager” type descriptive summaries and “IF-THEN” type associative rules [65], [76]. Typically linguistic summaries are extracted by generating all the possible rules and iterating over all the data points to generate quality measures for each rule. This can be highly time consuming, especially for highly multi-dimensional large datasets. This method is also susceptible to noise in the input data.

This chapter, presents a weighted Self-Organizing Map (SOM) based Linguistic Summarization method to derive “Yager” type and “IF-THEN” type linguistic summaries. The presented SOM-LS method uses the data compression, approximation, and generalization capabilities of the SOM to reduce the time complexity of the linguistic summarization process as well as providing a robust methodology that is transparent to noise that exists in data. The chapter also presents two new quality measures as well as modified quality measures that can be used to summarize data using SOM.
4.1 Linguistic Summarization of Data

In data mining applications it is important to understand the underlying interdependencies and characteristics of data in order to increase the understandability of a dataset. However, because of the advancement in information gathering and recording techniques, the amount of data available for various different applications is increasing fast, thus leading to a “data rich-information poor” paradigm [65], [76]-[78]. Therefore knowledge extraction from raw data in a human understandable form is becoming increasingly important for data mining. As the size of datasets increase rapidly, traditional, manual knowledge extraction is becoming impractical and obsolete [79]. One such method of knowledge extraction is data summarization. Data summarization can be defined as the process of extracting the most important information from a dataset to produce a concise, more understandable version for a particular application [76]-[80].

The most common and conventional method of data summarization is statistical or numerical summarization. Numerical summarization involves calculating parameters such as mean, median, standard deviation and other statistical quantities [80]. However, the problem with such summarization techniques is that they deal with terse, precise numbers, which is counterintuitive to human nature and natural language [80], [81]. Furthermore, such summarization may not be sufficient in complex scenarios. As shown in previous work on linguistic summarization, humans tend to better understand linguistically expressed properties rather than crisp exact values [65], [81]. Humans also prefer to exchange knowledge by means of linguistic information, even if it is less precise than numbers [80], [82], [83]. Furthermore, for some applications and for a higher level understanding of data, these precise values are not ideal [79], [80]. Linguistic summaries also tend to cope with non-numeric data better [65].
Linguistic summarization was first introduced by Yager in [82]. Since then, it has been used for various knowledge extraction applications and have been proven extremely useful and versatile in summarizing large multi-dimensional datasets [65], [76], [78], [84]-[86], and time series data [77], [79], [87]-[90]. Another area closely related to linguistic summarization is rule generation which is used for clustering and classification purposes [81], [91]-[94]. Rule generation deals with deriving predictive summaries, whereas linguistic summarization derives descriptive summaries [76]. Thus, this chapter focuses on generating descriptive and associative summaries which are more important in understandability of data during data pre-exploration and knowledge extraction from previously unseen, large datasets.

The most common type of linguistic summaries that is used is “Yager” type descriptive summaries which were introduced by Yager [82]. Yager type linguistic summaries take the form “Q y are S” where Q is a quantifier, y is a set of objects in the dataset and S is an output property which is a dimension of the dataset. For example “most students are undergraduates” is a Yager type summary where most is the quantifier, students are the set of objects and undergraduates is a property of the object. These types of summaries are an extension of dispositions proposed by Zadeh in [95], where Q is represented as a fuzzy quantifier. Yager type summaries are capable of extracting overall knowledge about objects in the database.

“IF-THEN” type association rules can also be used to summarize datasets [65], [76]. These summaries can take the form of fuzzy associative rules [1]. For example “IF A is SMALL THEN B is LARGE”, where A and B are the objects and SMALL and LARGE are fuzzy quantifiers. IF-THEN type linguistic summaries have been previously used for classification [91]-[94] and summarization of data [65], [76], [86], [96], [97]. Furthermore the same principles can be applied to fuzzy associative rule mining for the generation of fuzzy systems.
IF-THEN type summaries are capable of extracting causal relationships between dimensions of the dataset.

The conventional method of summarizing large datasets using linguistic summaries is iterating over the complete dataset, one data point at a time, for all possible summaries in order to verify the validity of generated summaries [76]. This exhaustive method is able to convey the distribution of a dataset. However, this exhaustive method is highly time consuming and can be susceptible to noise that is present in the data. There are many methodologies introduced in literature that reduces the overall time complexity of the linguistic summarization process [86], [92], [99], [100]. However, these techniques can be used in conjunction with the presented SOM-LS method, thereby gaining the same time advantage the exhaustive method does. Therefore, in this chapter the summarization time of the SOM-LS method is compared to the exhaustive method.

SOM have been widely used in the data mining field. Previous studies in predictive rule generation from data show the advantages of using SOM for linguistic data mining purposes. The main advantage of using SOM compared to other algorithms is its unsupervised learning capability. Furthermore, fast convergence, high robustness to noise, generalization capability and the ability to retain the relevant knowledge about the original dataset are also important factors in selecting SOM over other such algorithms. Many studies use the clustering capability of SOM to cluster data in a supervised or an unsupervised manner and generate predictive rules that describe these clusters [101]-[112]. Wong et al. uses the clustering capability of the SOM to cluster the dataset based on a set dimension and derive rules explaining the clusters [113]. Kaedi et al. uses SOM to cluster the generated rules and derive optimal rules based on these clusters [114].
In this chapter, linguistic summaries of data are derived using Type-1 Fuzzy Sets (T1 FSs) introduced by Zadeh in [1] and described in detail in Section 2.1.1. Therefore, in order to derive the summaries, first each dimension of the dataset needs to be fuzzified.

A dataset $D$ containing $M$ data points can be expressed as:

$$D = \{d_1, d_2, \ldots, d_M\}$$  \hspace{1cm} (4.1)

where $d_m$ represents a single data point and each data point $d_m$ has $N$ dimensions and is expressed as:

$$d_m = \{v_{m,1}, v_{m,2}, \ldots, v_{m,N}\}$$  \hspace{1cm} (4.2)

where $v_{m,n}$ is the $n^{th}$ dimension of the $m^{th}$ data point.

Each dimension is then fuzzified into a preset number of fuzzy sets. For the sake of simplicity, in this chapter, each dimension was fuzzified into three fuzzy sets as shown in Figure 8 (a). However, the range, the number and the type of fuzzy sets used can be predefined as desired by the user to increase or decrease the precision and understandability of the derived
rules. The higher the number of fuzzy sets used, the more descriptive the summaries are. However, the selection of fuzzy sets can be context dependent and an increase in the number of fuzzy sets also reduces the understandability of the derived summaries [81]. To increase the understandability of \textit{Yager} type summaries, the labels of each fuzzy set may vary according to the dimension.

4.1.1 \textbf{Yager-Type Linguistic Summarization}

A simple \textit{Yager} type linguistic summary can be expressed as:

\begin{equation}
Q \ y \ are \ S
\end{equation}

where $Q$ is the quantifier that specifies the amount of data points that follow the summary, $y$ denotes a set of objects that is described by the summary. $S$ is the output property of the summary and is a dimension in the dataset.

The \textit{Yager} type summary shown in (4.3) can be written in terms of dimensions and their corresponding fuzzy sets as:

\begin{equation}
S(Q) \ y \ are \ S(v_c)
\end{equation}

where $v_c$ is a dimension of the dataset. $S(v_c)$ is the fuzzy set of $v_c$, and $S(Q)$ is the fuzzy set of the quantifier. In this chapter, the quantity fuzzy sets, $S(Q)$ were generated using the fuzzy sets shown in Figure 8 (b). A more complex and descriptive \textit{Yager} summary follows the form:

\begin{equation}
Q \ R \ y \ are \ S
\end{equation}
where $R$ is a qualifier, which is a dimension of the dataset and is represented by a fuzzy set. Multiple qualifiers may be used to increase the complexity and accuracy of summaries. However, the increase in the number of qualifiers also decreases the understandability of the summary. Thus a complex Yager type summary with two qualifiers can be written in terms of dimensions and their corresponding fuzzy sets as:

$$S(Q) \ S_1(v_a) \ S_2(v_b) \ y \ are \ S_3(v_c)$$

(4.6)

where $a \neq b \neq c$ and $v_a$, $v_b$ and $v_c$ are dimensions of the dataset. $S_i$ are the fuzzy sets for each dimension.

### 4.1.2 IF-THEN-Type Linguistic Summarization

An IF-THEN linguistic summary takes the same form as a T1 fuzzy rule described in Section 2.1.3, and formally represented in (2.19). This fuzzy rule can be re-written to be analogous to the form shown in (4.6) as:

$$IF \ v_a IS S_1 AND v_b IS S_2 THEN v_c IS S_3$$

(4.7)

where, as before, $a \neq b \neq c$ and $v_a$, $v_b$ and $v_c$ are the dimensions of the dataset. $S_i$ are the fuzzy sets for each dimension.

### 4.2 Quality Measures for Linguistic Summarization of Data

Many quality measures and interestingness measures that can rank a given summary for a dataset can be found in literature [65], [76], [80], [82], [85], [93], [96], [115]-[117]. For
ranking Yager type linguistic summaries, the degree of truth as presented in many previous works is used in this chapter [78], [82], [87], [117]. For ranking IF-THEN type summaries, three recently proposed quality measures are used: degree of truth, degree of sufficient coverage and degree of reliability which were proposed by Wu et al. in [65], [76]. In addition to these quality measures, two new quality measures that can be used to rank both Yager type summaries and IF-THEN type summaries are presented: normalized cardinality and optimized reliability. These novel quality measures take into account the number of data points a summary covers, along with the membership degree of the covered data points. Each presented quality measure has advantages and disadvantages and none can be used by itself to generate optimal summaries for a dataset [117]. However, each quality measure is able to produce a value between 0 and 1 that can be used to derive the most important summaries that are persistent in the dataset.

**Degree of Truth:** The degree of truth was first proposed by Yager for linguistic summarization in [81], and is an extension of the truth degree used by Zadeh in [95] for fuzzy dispositions. The degree of truth is calculated differently for basic Yager type summaries with no qualifiers and more complex summaries with one or more qualifiers. It is calculated by deriving the t-norm of the membership degree for each qualifier and output property of the summary, for each data point.

Thus the degree of truth for a basic Yager type summary shown in (4.4) can be calculated using:
where $M$ is the total number of data points, $\mu_{S(Q)}$ is the membership degree of the quantity $Q$, and $\mu_{S_1(v_m,c)}$ is the membership degree of value $V$, of data point $m$, in dimension $C$ for fuzzy set $S_1$.

For the more complex Yager type summary shown in (4.6) the degree of truth is calculated as:

$$T_{Yager} = \mu_{S(Q)} \left[ \sum_{m=1}^{M} \frac{\mu_{S_1(v_m,c)}}{M} \right]$$ (4.8)

where $\Pi$ is the fuzzy t-norm operation. In both Equations (4.8) and (4.9), the t-norm of the membership degree that contributes to the summary is calculated for each data point. The use of the quantity set ensures the number of data points that follow each summary is taken into account.

Similar to Yager type summaries, the degree of truth for IF-THEN type summaries is calculated by deriving the t-norm of the membership degree for each antecedent and consequent, for each data point [65], [76]. Thus the degree of truth for the summary in (4.7) is expressed as:
\[
T_{IF-THEN} = \frac{\sum_{m=1}^{M} \mu_{S_1}(v_{m,a}) \prod \mu_{S_2}(v_{m,b}) \prod \mu_{S_3}(v_{m,c})}{\sum_{m=1}^{M} \mu_{S_1}(v_{m,a}) \prod \mu_{S_2}(v_{m,b})}
\]  

(4.10)

As the number of data points that does not satisfy the consequent increase the degree of truth is decreased. As the number of data points satisfy the antecedents is decreased the denominator goes to zero. Therefore, the larger the degree of truth the more data points among those satisfying antecedents also satisfy consequents. However the degree of truth as used for \textit{IF-THEN} type summaries does not elaborate on the number of data points each summary is associated with. Thus, the degree of sufficient coverage is used in-conjunction with the degree of truth for \textit{IF-THEN} summaries.

**Degree of Sufficient Coverage:** The degree of sufficient coverage was presented by Wu et al. [1], [2], and is calculated using the percentage of data points that supports a certain \textit{IF-THEN} type summary. Thus degree of sufficient coverage \( C_{IF-THEN} \) is expressed as:

\[
C_{IF-THEN} = f_c \left( \sum_{m=1}^{M} t_m \right)
\]

(4.11)

where

\[
t_m = \begin{cases} 
1, & \mu_{S_1}(v_{m,a}) \prod \mu_{S_2}(v_{m,b}) \prod \mu_{S_3}(v_{m,c}) > 0 \\
0, & \text{otherwise}
\end{cases}
\]

(4.12)

The function \( f_c \) is the sigmoid function, as shown in Figure 9, that maps the ratio of data points that supports the rule to a value between 0 and 1. The values \( r_{\text{min}} \) and \( r_{\text{max}} \) can be set.
by the user according to prior knowledge of the data distribution, thus, this ability enables users to specify the sufficient coverage required by a generated summary.

**Degree of Reliability**: Degree of reliability, presented in [76] combines the degree of truth and degree of sufficient coverage. An *IF-THEN* type summary is more expressive of the data if it has a high truth and a high coverage. In other words if either the degree of truth or the degree of sufficient coverage is low, the summary would not contain important knowledge about the dataset. Thus degree of reliability is expressed as:

$$R_{IF-THEN} = \min(T_{IF-THEN}, C_{IF-THEN})$$  \hspace{1cm} (4.13)

**Normalized Cardinality**: The first novel quality measure presented in this manuscript is normalized cardinality. Normalized cardinality is a quality measure based on both the degree of truth and degree of coverage. Normalized cardinality $N_C$ is calculated by:

$$N_C = \frac{\sum_{m=1}^{M} \mu_{S_1}(v_{m,a}) \prod \mu_{S_2}(v_{m,b}) \prod \mu_{S_3}(v_{m,c})}{M}$$  \hspace{1cm} (4.14)

Figure 9. Sigmoid function used to calculate the degree of sufficient coverage
According to the equation the normalized cardinality is decreased as the number of data points that supports a summary is decreased. If either a data point does not match the antecedent or the consequent of the summary, the normalized cardinality is reduced. Since each data point is considered in the calculation, the coverage of a summary is also included in the normalized cardinality. Thus normalized cardinality encodes the number of data points a summary covers and captures the amount of which each data point follows a summary.

**Optimized Reliability**: The second novel quality measure presented is the optimized reliability. Optimized reliability further enhances the usability of normalized cardinality and includes the validity of a generated summary. The calculated normalized cardinality is multiplied by the degree of truth to generate optimized reliability:

\[
O_R = N_C \times T
\]

Thus summaries that have a high degree of truth and a high normalized cardinality will have a high optimized reliability. By using both degree of truth and normalized cardinality the optimized reliability provides a combination of the validity of the summary along with the amount of data that follows the summary. Therefore, optimized reliability provides a similar measure of summaries to that of degree of reliability.

### 4.3 Linguistic Summarization of Data Using Self-Organizing Maps (SOM-LS)

This section presents SOM-LS method, which is a SOM-based, weighted, linguistic summarization method that derives Yager type and *IF-THEN* type fuzzy linguistic summaries. SOM was used for its ability to generalize and approximate data using a fixed number of neurons.
The presented SOM-LS method identifies each neuron in the SOM as a data point, thereby compressing the number of data points and increasing the efficiency of the summarization process. Thus, the presented method uses the weight vectors of the SOM neurons as a generalized representation of data. This facilitates faster computation time without a reduction in quality of the generated summaries. Because neurons are a generalized representation of data, using the weight vectors of the neurons also enable more robust summary generation. However, due to the data compression, some neurons are associated with more data points than others, thus increasing the importance of some neurons for the generation of summaries. Therefore it is important to emphasize the number of actual data points when calculating the quality measures using SOM. The number of times a neuron was selected as the best matching unit ($N_{NBU,K}$ see Equation (4.20)) was used to emphasize neurons that encode more data, and at the same time eliminate the neurons that does not encode any data.

4.3.1 Self-Organizing Maps (SOM)

The Self-Organizing Map (SOM) algorithm was developed in 1981 [118] and is frequently used in data mining as dimensionality reduction and data compression tools. SOM uses unsupervised winner-takes-all competitive learning method together with cooperative adaptation to adjust itself to the topological properties of the input dataset. The SOM consists of a topological grid of neurons typically arranged in 1D, 2D or 3D lattice [119], [120]. Each neuron has a position in input space as well as a fixed position in output space. The fixed grid defines the spatial neighborhood of each neuron.
Each neuron maintains a synaptic weight vector \( \hat{\mathbf{w}} = \{w_1, \ldots, w_N\} \), where \( N \) is the dimensionality of the input space. The input dataset consists of input patterns that can be denoted as \( \tilde{d}_m = \{v_{m,1}, v_{m,2}, \ldots, v_{m,N}\} \). The training process can be described in several steps as follows [119]:

**Step 1: Initialization:** Randomly initialize all synaptic weight vectors in the input domain.

**Step 2: Sampling:** Select a random input pattern \( \tilde{d}_i \) from the training dataset.

**Step 3: Competitive Learning:** Find the Best Matching Unit (BMU) for the current input pattern \( \tilde{d}_i \). The BMU is found by minimizing the Euclidean distance between the input pattern \( \tilde{d}_i \) and the synaptic weight vectors \( \hat{w} \):

\[
BMU(\tilde{d}_i) = \arg \min_k \| \tilde{d}_i - \hat{w}_k \|, \quad k = 1, 2, \ldots, K
\]  

(4.16)

where \( BMU(\tilde{d}_i) \) is the best matching unit for input pattern, operator \( \| \| \) denotes the Euclidian distance norm, and \( K \) is the set of all the neurons in the SOM.

**Step 4: Cooperative Updating:** Update the synaptic weight vectors of all neurons in SOM using the cooperative update rule:

\[
\hat{w}_k(j + 1) = \hat{w}_k(j) + \eta(j) h_{k, BMU(\tilde{d}_i)}(j) (\tilde{d}_i - \hat{w}_k(j))
\]  

(4.17)

Here, \( j \) denotes the iteration, \( \eta(j) \) is the learning rate and \( h_{k, BMU(\tilde{d}_i)}(j) \) is the value of the neighborhood function for the neuron \( k \) centered at \( BMU(\tilde{d}_i) \).
**Step 5: Convergence Test:** Until a specified convergence criterion is met, it is important to repeat from Step 2.

Here, the learning process is controlled by the dynamic learning rate and the neighborhood function. The neighborhood function is typically implemented as a Gaussian function centered at the selected winning neuron. Its amplitude applied to neuron \( k \) can be calculated as follows:

\[
h_k,\text{BMU}(d_j) = \exp \left( -\frac{\| r_k - r_{\text{BMU}(d_j)} \|^2}{2\sigma^2} \right)
\]

where \( r \) is the two dimensional vector that represents the position of the neuron in the output space.

The size of the Gaussian neighborhood function is determined by parameter \( \sigma \). In order to enforce a convergent behavior, the size of neighborhood is reduced by decreasing the parameter \( \sigma \). Typically, the exponential decay rule is applied. The learning rate \( \eta \) controls the rate of adaptation of individual neurons. Like the size of the neighborhood function, its value also exponentially decays with the elapsed training time.

The learning process described in Steps 2-5 is repeated until a certain criterion is met. In this dissertation the following specific convergence criterion was used:

\[
\sum_{k=1}^{K} \frac{\sum_j \text{abs}(\tilde{w}_k(j+1) - \tilde{w}_k(j))}{K} \geq \delta
\]  

(4.19)
This criterion is defined as the average weight change of all the neurons after every training cycle. Once the average weight change drops below the predefined constant $\delta$, the training is terminated. The constant $\delta$ is typically set to a small value (around $10^{-5}$).

For this specific application, the number of times each neuron $k$ was selected as a best matching unit (BMU) was stored as $N_{NBU,k}$, such that:

$$
\sum_{k=1}^{K} N_{BMU,k} = M
$$

(4.20)

where $K$ is the number of neurons and $M$ is the total number of data points. This value was calculated after the convergence of the SOM.

### 4.3.2 Quality Measures for Linguistic Summarization of Data Using SOM

To enable data representation using the generalized neurons of the SOM, each quality measure described in Section 4.2 is modified. This modification is done to facilitate the inclusion of information about the number of data points that support each summary when used with SOM neurons.

**Degree of Truth:** In order to calculate the degree of truth using the SOM-LS method, the membership degree is calculated for the weight vector of the neurons. Thus Equations (4.8) and (4.9) were modified to express the SOM based degree of truth as:

$$
TS_{Yager} = \mu_{S(Q)} \left[ \frac{\sum_{k=1}^{K} \left( \mu_{S1}(w_{k,c}) \times N_{BMU,k} \right)}{M} \right]
$$

(4.21)
\[ T_{S\_ager} = \mu_{S(Q)} \left[ \frac{\sum_{k=1}^{K} \mu_{S_1}(w_{k,a}) \prod_{k} \mu_{S_2}(w_{k,b}) \prod_{k} \mu_{S_3}(w_{k,c})}{\sum_{k=1}^{K} \mu_{S_1}(w_{k,a}) \prod_{k} \mu_{S_2}(w_{k,b})} \right] \]  

(4.22)

where \( w_{k,i} \) is the weight of neuron \( k \) for dimension \( i \), and \( N_{NBU,K} \) is the number of times neuron \( k \) was selected as the best matching unit. Similarly, Equation (4.10) can be modified to express the SOM based degree of truth as [66]:

\[ T_{S\_IF\_THEN} = \frac{\sum_{k=1}^{K} \mu_{S_1}(w_{k,a}) \prod_{k} \mu_{S_2}(w_{k,b}) \prod_{k} \mu_{S_3}(w_{k,c})}{\sum_{k=1}^{K} \mu_{S_1}(w_{k,a}) \prod_{k} \mu_{S_2}(w_{k,b})} \]  

(4.23)

**Degree of Sufficient Coverage**: As before, the degree of sufficient coverage expressed in (4.11) can be modified as [66]:

\[ CS_{S\_IF\_THEN} = f_{c} \left( \sum_{k=1}^{K} \left( t_{k} \times N_{BMU,k} \right) \right) \]  

(4.24)

where

\[ t_{k} = \begin{cases} 1, & \mu_{S_1}(w_{k,a}) \prod_{k} \mu_{S_2}(w_{k,b}) \prod_{k} \mu_{S_3}(w_{k,c}) > 0 \\ 0, & otherwise \end{cases} \]  

(4.25)

The coverage is calculated using the weight vectors of the SOM; however, the number of data points is included in the calculation by means of \( N_{NBU,K} \).

**Degree of Reliability**: This quality measure can be derived from equation (4.13) and can be expressed as [66]:

71
\[ RS_{IF-THEN} = \min(TS_{IF-THEN}, CS_{IF-THEN}) \]  

**Normalized Cardinality:** Similar to degree of sufficient coverage, normalized cardinality takes into account the number of data points associated with a summary. Thus, the number of times each neuron is selected as a best matching unit is used to modify equation (4.14) as:

\[ NS_C = \frac{\sum_{m=1}^{M} (\mu_{S1}(v_{m,a}) \prod \mu_{S2}(v_{m,b}) \prod \mu_{S3}(v_{m,c})) \times N_{NBU,K}}{M} \]  

**Optimized Reliability:** SOM-based optimized reliability is calculated using SOM-based degree of truth, \( TC \) and SOM-based normalized cardinality, \( NS_C \) as:

\[ OS_R = NS_C \times TS \]  

Thus, these modified quality measures can be used with a trained SOM rather than the original dataset to obtain linguistic summaries of the original data.

### 4.4 Experimental Results

The presented SOM-LS method for linguistic summarization is compared to conventional exhaustive linguistic summarization in this section. SOM-LS method was tested and verified using one artificial dataset and four real-world benchmark datasets. The artificial dataset was 10-dimensional, and contained 5 Gaussian clusters with random cluster radius and random centers of gravity. The four real-world datasets selected were *Auto MPG, Blood Transfusion, Pima Indian Diabetes* and *Wine Quality* datasets [147]. These datasets are summarized in Table 3.
The selected benchmark datasets contained two classification and two regression problems, as well as two low complexity and two high complexity problems. Auto MPG and Blood Transfusion datasets were considered low complexity datasets because of low number of data points and low dimensionality, while Wine Quality and Pima Indian datasets contained higher dimensions and higher number of data points, and thus were considered to be of relatively higher complexity.

In this section, first, the computational complexity of the presented SOM-LS method compared to exhaustive method is analyzed. This is then empirically demonstrated by generating summaries for different sized datasets. Second, in order to evaluate the quality of the summaries generated by the SOM-LS method, the summaries generated by the SOM-LS method are compared with summaries generated by the exhaustive method. Although in some cases the summaries generated using the exhaustive method may not be optimal, exhaustive method was used as the benchmark. Finally, the robustness of the proposed method is shown by introducing different levels of noise into the original data and generating summaries. The generated summaries were then compared with the summaries generated with original non-noisy data to evaluate the performance of the two methods in noisy conditions.

Table 3. Benchmark datasets used for evaluation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Number of Dimensions</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto MPG</td>
<td>398</td>
<td>8</td>
<td>Regression</td>
</tr>
<tr>
<td>Blood Transfusion</td>
<td>748</td>
<td>5</td>
<td>Classification</td>
</tr>
<tr>
<td>Pima Indians Diabetes</td>
<td>768</td>
<td>8</td>
<td>Classification</td>
</tr>
<tr>
<td>Wine Quality</td>
<td>4898</td>
<td>12</td>
<td>Regression</td>
</tr>
</tbody>
</table>
4.4.1 Run Time Analysis

The basic linguistic summarization methodology employs an exhaustive search whereby all possible summaries that can be generated are evaluated using the given dataset. Thus, the overall run time of the algorithms depends on the number of possible summaries and the number of data points in the dataset. As the dimensionality of the dataset and complexity of the summaries increase, it presents a combinatorial problem because a large number of summaries must be generated and validated [148], [149]. Thus the time complexity of the summary generation process depends on the dimensionality of the dataset as well as the required complexity of the summaries. Assuming each dimension is fuzzified into equal number of fuzzy sets, the number of Yager type summaries generated for a data set with \( N \) number of dimensions can be calculated as:

\[
R_{Yager} = P_Q \left[ PN \sum_{i=1}^{Z} \left( \begin{array}{c} N-1 \\ i \end{array} \right) P^i \right]
\]  

(4.29)

where \( P \) is the number of fuzzy sets for each dimension, \( P_Q \) is the number of quantifier fuzzy sets, \( Z \) is the number of qualifiers and, \( \left( \begin{array}{c} N-1 \\ i \end{array} \right) \) represents all possible combinations of \( N-1 \) dimensions for \( i \) qualifiers.

Similarly the number of summaries generated for IF-THEN type summaries \( R_{IF-THEN} \) can be calculated using:

\[
R_{IF-THEN} = PN \sum_{i=1}^{Z} \left( \begin{array}{c} N-1 \\ i \end{array} \right) P^i
\]  

(4.30)

where, \( P \) is the number of fuzzy sets for each dimension and \( Z \) is the number of antecedents. Thus using the number of summaries generated, \( R \), the time complexity for generating
summaries using the exhaustive method is \( O(RM) \). Similarly, the time complexity for generating summaries using SOM-LS method is \( O(RK) \). However, the SOM-LS method also requires SOM training time. The SOM training time depends on the number of data points: \( M \), the dimensionality of the dataset: \( N \), the number of neurons in the SOM: \( K \), and the number of training iterations: \( e \). It has been shown that the time complexity of SOM training is \( O(NMKe) \) [149]. Thus the overall time complexity for the SOM-LS method is \( O(\max(NMKe, RK)) \). Therefore, when the number of data points is significantly larger and the dimensionality of the data is higher, the run time of the SOM-LS method is asymptotically bound by the SOM train time \( O(NMKe) \). In such cases, the presented method has a better time complexity compared to the exhaustive method when \( R > NKe \). Thus, with large datasets the presented method will have better time complexity. Furthermore, since the dimensionality of the data is known, using the above inequality, a suitable size of SOM can be selected such that the time complexity advantage is retained.

In order to evaluate the gained time complexity advantage of summarization of data using the presented SOM-LS method, summaries were generated for different sized datasets using SOM-LS and exhaustive methods separately. The computation time for each approach was recorded over multiple runs and averaged to produce the final run time. A set of 10 dimensional artificial datasets, each containing 5 clusters were used for this comparison. 50 datasets were created, each containing \( M \) data points such that:

\[
M_{i+1} = M_i + 25 \quad i = 1, 2, \ldots, 49
\]  

(4.31)
where \( M_i \) is the number of data points in the current dataset, and \( M_1 \) was set to 100. Run time for each method was calculated by generating \( Yager \) type and \( IF-THEN \) type summaries 10 times for each dataset and averaging final value to generate the run time for each method.
The exhaustive method was compared with four different SOM sizes: 5×5 square grid, 10×10 square grid, 15×15 square grid, and 20×20 square grid. The final run time includes the training time of SOM.

Figure 10 plots the Yager type and IF-THEN type run times for each dataset, respectively. Clearly the run-time of the exhaustive method is linearly dependent on the number of data points in the dataset, thus increasing the time complexity of the process. Since the size of the SOM remains the same, the run time for the summary generation process using SOM is independent from the number of data points.

It has to be noted that other techniques that are aimed at reducing the time complexity of the linguistic summarization process are mostly heuristic search techniques that can be applied along with the presented SOM-LS method that can leverage the advantages of both methods. Furthermore, for a dataset that is continuously growing, re-evaluation of summaries will be much faster in the SOM-LS method.

### 4.4.2 Comparison of Generated Summaries

The quality of summaries generated using SOM-LS method is validated in this section by comparing summaries with highest quality rating generated by the SOM-LS method and the exhaustive method. For this analysis, Yager type summaries containing 0 to 3 qualifiers and IF-THEN type summaries containing 1 to 3 antecedents were generated. Different sized SOM containing 25 to 400 neurons arranged in a square grid were used to generate summaries to illustrate the effect of the size of the SOM on the summary generation process.
Top 10 summaries (according to quality rating) for both types were generated using the two methods, and compared. Table 4 to Table 7 show the number of matching summaries in the SOM-LS method and exhaustive method for the four benchmark datasets illustrated in Table 3. Table 4 shows the results for Auto MPG dataset. Small 5×5 SOM was unable to provide a sufficient approximation of the dataset. Hence, 5×5 SOM had a low number of
matching summaries. However with larger SOM the number of matching summaries increased. The 20×20 SOM contained 400 neurons, which is larger than the dataset, therefore, the run time for 20×20 SOM was higher than exhaustive method.

Table 6. Comparison between the summaries generated by the Exhaustive method and the SOM method for Pima Indian Diabetes dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Yager type</th>
<th>IF-THEN type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Matches with Exhaustive top 10</td>
<td>Run time (s)</td>
</tr>
<tr>
<td>5×5 SOM</td>
<td>5</td>
<td>1.76</td>
</tr>
<tr>
<td>10×10 SOM</td>
<td>6</td>
<td>2.81</td>
</tr>
<tr>
<td>15×15 SOM</td>
<td>8</td>
<td>4.98</td>
</tr>
<tr>
<td>20×20 SOM</td>
<td>10</td>
<td>7.72</td>
</tr>
</tbody>
</table>

Table 7. Comparison between the summaries generated by the Exhaustive method and the SOM method for Wine Quality dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Yager type</th>
<th>IF-THEN type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Matches with Exhaustive top 10</td>
<td>Run time (s)</td>
</tr>
<tr>
<td>5×5 SOM</td>
<td>4</td>
<td>9.55</td>
</tr>
<tr>
<td>10×10 SOM</td>
<td>4</td>
<td>16.03</td>
</tr>
<tr>
<td>15×15 SOM</td>
<td>10</td>
<td>29.36</td>
</tr>
<tr>
<td>20×20 SOM</td>
<td>10</td>
<td>47.27</td>
</tr>
</tbody>
</table>
Results for the Blood Transfusion dataset are shown in Table 5. Even though Blood Transfusion dataset contained more data points than Auto MPG the run time was considerably low. This is due to the lower dimensionality of the Blood Transfusion dataset.

Pima Indian dataset (shown in Table 6) is similar in size to Blood Transfusion dataset. However, the number of dimensions is higher and thus the run time higher in comparison. Finally the same analysis was performed for Wine Quality dataset and the results are shown in Table 7.

These results show that while $5 \times 5$ and $10 \times 10$ SOM provides better run times, the number of neurons is insufficient to generalize large multi-dimensional datasets accurately, thus diverging from the expected result. However, larger sized SOM provides a more accurate representation of the data while maintaining the time advantage over the exhaustive method. Thus it can be concluded that using an appropriate size SOM allows the generation summaries without affecting the quality of the summaries and with a better time complexity than the exhaustive method.

4.4.3 Robustness to Noise

The robustness analysis shows the robustness of the SOM-LS method to noise that can be present in the input data. The robustness of linguistic summarization process was evaluated by adding increasing amounts of noise to the original data and comparing the generated summaries with the summaries generated using non-noisy data.
For this analysis, noise from Signal-to-Noise Ratio (SNR) 20dB to SNR 10dB was introduced to the original data (see equation 3.38). Using these noisy data, top 10 summaries were generated using both the SOM-LS method and the exhaustive method. The generated summaries were then compared with the top 10 summaries generated from the original data.

Figure 11. Quality degradation of generated “Yager” type summaries at different noise levels for each dataset (a) Auto MPG (b) Blood Transfusion (c) Pima Indian Diabetes, and (d) Wine Quality

For this analysis, noise from Signal-to-Noise Ratio (SNR) 20dB to SNR 10dB was introduced to the original data (see equation 3.38). Using these noisy data, top 10 summaries were generated using both the SOM-LS method and the exhaustive method. The generated summaries were then compared with the top 10 summaries generated from the original data.
Figure 11 plots the deterioration of the quality of the generated Yager type summaries with increasing amounts of noise for the tested datasets. Similarly, Figure 12 plot the same information for IF-THEN type summaries.

Figure 12. Quality degradation of generated "IF-THEN" type summaries at different noise levels for each dataset (a) Auto MPG (b) Blood Transfusion (c) Pima Indian Diabetes, and (d) Wine Quality.

For the robustness analysis the two SOM architectures that performed best in sub section 4.4.2., i.e. 15×15 and 20×20 were used. However, for the Auto MPG dataset only
the 15×15 SOM was used because the number of data points in the Auto MPG dataset is lower than the number of neurons in a 20×20 grid.

For all benchmark datasets tested, the SOM-LS method was observed to be more transparent to noisy data. For all datasets, the smaller 15×15 SOM provides slightly better robustness. This is because a smaller sized SOM provides higher generalization of the input space compared to a larger sized SOM. Therefore it is crucial to select the size of SOM that provide sufficient approximation such that the quality of the summaries is not affected, and generalization such that the summaries will be sufficiently transparent to noise. The robustness analysis illustrates the generalization capability of the presented SOM-LS method. Thus, SOM-LS method is more transparent to noise and generates summaries that are more robust.

4.5 Conclusions

This chapter presented a Self-Organizing Map based Linguistic Summarization method (SOM-LS), which is a methodology for Linguistic Summarization of data by using the data compression, approximation and generalization capabilities of Self-Organizing Maps. The presented method can be utilized to generate either Yager type of IF-THEN type fuzzy linguistic summaries that summarizes datasets in a human understandable format. Two new quality measures, the Normalized Cardinality and the Optimized reliability that encode the amount of data as well as the validity of the summary were introduced in this chapter. Methodology to adopt these quality measures as well as other quality measures found in literature to be used with the presented SOM-LS method was also presented in this chapter.
The computational complexity of the presented method was evaluated showing the advantages as well as limitations of the SOM-LS method. The computational complexity analysis also yields upper bounds on the SOM size that will result in a run time advantage over the conventional method. The presented SOM-LS method was compared to the conventional exhaustive method using different benchmark and artificial datasets. The computational advantage and the robustness to noise as well as the capability of generating summaries high quality summaries of the presented method was shown in this comparison.

It has to be noted that the presented method can be used in conjunction with other proposed methods in literature to further increase the computational efficiency as well as the quality of the generated summaries. Another significant advantage of the presented method is the capability of handling continuously increasing datasets. In such cases, the trained SOM acts as previous knowledge of data and therefore, the recalculation of summaries will entail less computational complexity.
CHAPTER 5

SHADOWED TYPE-2 FUZZY LOGIC SYSTEMS

General Type-2 Fuzzy Sets (GT2 FSs) were originally proposed by Lotfi Zadeh in [18] because Type-1 (T1) FSs may over-specify vague terms in some cases because of the real-valued membership grades. Thus GT2 FSs has the capability of modeling the uncertainty associated with the membership degree of T1 FSs [121]. General Type-2 Fuzzy Logic Systems (GT2 FLSs) are therefore an extension to T1 FLSs [122], differing in the nature of individual Fuzzy Sets (FSs), where GT2 FLSs use GT2 FSs to model the fuzzy rule antecedents and consequents. However, due to the high computational complexity of operations on GT2 FSs, GT2 FLSs have been rarely used in practical applications. Instead, Interval Type-2 (IT2) FLSs which employ constrained IT2 FSs, have been widely used. Despite their superior computational complexity, IT2 FLSs lack the expressive power of GT2 FSs when describing various sources of uncertainty. Further, it is unclear how to derive an IT2 FLS from a specific GT2 FLS.

To alleviate these issues, this chapter presents and outlines the design of Shadowed Type-2 Fuzzy Logic Systems (ST2 FLSs), which employ Shadowed Type-2 (ST2) FSs to model the fuzzy rule antecedents and consequents. ST2 FSs are Type-2 FSs with Shadowed Sets (SSs) representing their secondary membership functions. SSs were originally introduced by Pedrycz in 1998 [123]-[127].
The ST2 FLS design presented in this chapter is automatically derived from an original GT2 FLS in order to preserve the uncertainty modeled by the original GT2 FSs. ST2 FLSs can therefore offer improved modeling of uncertainty when compared to IT2 FLS while also providing efficient computational framework since the secondary membership grades can only have three values which are, 0, 1, or completely uncertain (shadowed) grade of [0, 1]. Experimental results show that the presented ST2 FLS models the more complex GT2 FLS more closely than an IT2 FLS and that ST2 FLSs are computationally efficient compared to GT2 FLS since they apply the efficient fuzzy inference mechanisms of IT2 FLSs.

5.1 Uncertainty Modeling

Uncertainty modeling, and propagation analysis is a crucial task for control systems and data mining applications alike [24], [128], [129]. The task of uncertainty modeling in complex systems is difficult and time consuming due to various interdependencies as well as different types of uncertainties present [24], [27]. While classification of uncertainties is difficult by nature, two different types of uncertainties are discussed in literature in terms of the modeling techniques involved: stochastic uncertainties and epistemic uncertainties [6], [20], [27], [128].

Uncertainties that arise due to inherent known variability that can be present in the system are known as stochastic uncertainties which are also referred to as aleatory uncertainties in the literature [128], [130]. The probability distributions of uncertainties are known in stochastic uncertainties [6], [130]. Therefore, well documented and powerful probabilistic models are present for modeling and characterizing these uncertainties as well as for analyzing
their propagation through the system [131]. In addition to the probability distribution of uncertainties, a deep understanding of the system is required to model the propagation of uncertainty [24], [25], [27]. Techniques such as Monte-Carlo simulations coupled with different probabilistic models are widely used in industry to model these uncertainties [24], [130], [131]. Examples of stochastic uncertainties are mostly categorized as noise that has a known distribution that can be present in measurements and data [24], [130], [131]. However, in most real world scenarios, the true distribution of the noise is not known [6], [28], [132]. Thus, while in many cases statistical modeling is accurate and sufficient, there are situations where statistical modeling cannot be utilized [6], [132].

The uncertainties that arise from insufficient or lack of complete knowledge of the system is known as epistemic uncertainties [128], [130], [131]. These type of uncertainties are difficult to model using classical mathematical and statistical models [6], [132]. However, it has been shown that Fuzzy Logic Systems are a powerful methodology for modeling epistemic uncertainties [6], [133]-[136]. Furthermore, fuzzy arithmetic can model how different epistemic uncertainties propagate through complex systems using expert knowledge in the form of linguistic rules [6], [133]. Thus, in situations where statistical probabilistic models fail to model uncertainties, fuzzy systems are used [130], [136]. Examples of these types of uncertainties include, random noise where the distribution of noise is unknown, and situations where the true interdependencies between systems are unknown or only partially unknown [6], [26]-[28]. Recent research have shown that General Type-2 Fuzzy Logic Systems are the most powerful and capable method of handling such uncertainties [6], [8], [10]. The most commonly used method to evaluate the uncertainty modeling capability of a FLS is to measure its performance in noisy situations [6], [8], [10].
5.2 Shadowed Type-2 Fuzzy Logic Systems

GT2 FSs use membership degrees that are themselves FSs. Despite the powerful uncertainty modeling capability of GT2 FSs, the high computational complexity of computing with GT2 FSs significantly hindered their practical use. As a consequence, GT2 FLSs have been rarely applied in practice [137]. Recently, there has been a renewed interest in the area of GT2 FSs and GT2 FLSs due to the recently introduced representations of geometric T2 FSs [44], [138] or the $\alpha$-planes [41]-[43] and the zSlices [40], [139] representations. The high computational complexity of GT2 FLSs led to a wide spread of applications of their constrained version which are the Interval T2 (IT2) FLSs [140], [141]. The IT2 FLSs consist of IT2 FSs which restrict the form of the secondary membership functions to intervals [142] [143]. This simplification allows the development of efficient algorithms for fuzzy inference with IT2 FSs [144]. However, the restricted interval secondary membership functions can be seen as a significant limitation in situations where more complex representation of secondary uncertainty is required [32], [6].

Hence, on one side there are GT2 FLSs with rich uncertainty modeling capability but with unfavorable computational complexity. On the other side there are IT2 FLSs which provide efficient computational framework but at the price of significantly restricting the options for modeling various sources of uncertainty [40]. This chapter presents a new class of FLSs, the Shadowed Type-2 Fuzzy Logic Systems (ST2 FLSs), which constitute a compromise of both methods.

The presented class of ST2 FLS is based on the concept of ST2 FSs [121]. An ST2 FS is a GT2 FS with all secondary membership functions represented as Shadowed Sets (SSs) [123]-[127]. The computational complexity of processing ST2 FSs is significantly reduced
because it is able to take advantage of the efficient fuzzy operations on IT2 FSs. However, at the same time, the ST2 FSs offer improved description of uncertainty, which is captured using the SSs rather than simple interval values for the secondary fuzzy membership functions. Similar representation to ST2 FSs, named Shadowed Fuzzy Sets was recently outlined in [145], [146].

5.2.1 Shadowed Type-2 Fuzzy Sets (ST2 FSs)

This section provides a detailed overview of the concept of ST2 FSs which was presented in [19], [121]. The concept of Shadowed Sets (SSs) was originally developed to improve the observability and interpretability of T1 FSs and to alleviate the issues of excessive precision in describing imprecise concepts when using T1 fuzzy membership functions [123]-[125]. Meaning, that in some cases the real valued membership degree may be too precise in expressing the uncertainty in vague linguistic terms that describe real-world phenomena [18].

The primary driver behind the concept of SSs is that the values close to membership degrees 0 and 1 can be easily described. This means that people can easily come to a consensus when a term is exactly certain or exactly uncertain. For example, people can agree when it is absolutely “too cold” or absolutely “too hot”, but not on values in between. But assigning other membership degrees is difficult and people often cannot come to a consensus [121], [123]. A SS is directly induced by a T1 FS. Based on the T1 fuzzy membership grades, the SS can be divided into three regions: exclusion, core and shadow [123]-[125].
5.2.1.1 **Shadowed Sets (SSs)**

Figure 13 depicts a T1 FS $A$ in the input domain $X$ and is defined using the MF $\mu_A(x)$ for $\forall x \in X$. The T1 FS, $A$ can be observed as a functional mapping of the input domain $X$ into the interval $[0, 1]$ [123], [121]:

$$A : X \rightarrow [0, 1]$$  \hspace{1cm} (5.1)

The T1 FS $A$ directly induces a SS $\overline{A}$ by converting the functional mapping in (5.1) into a three valued logic that represents the above mentioned exclusion, core and shadow regions. However, the uncertainty associated with the initial T1 FS, $A$, must be encapsulated in the derive SS $\overline{A}$. This is done by three tasks: *elevating*, *reducing* and *balancing* [123], [121].

Figure 13. Type-1 fuzzy set and the induced shadowed set depicting the three regions of the shadowed set (Core, Shadow, and Exclusion)
A threshold $\lambda$ is applied to the T1 MF $\mu_A(x)$ in order to generate the SS, $\tilde{A}$. The membership degrees above $1-\lambda$ in $A$ are elevated to 1, creating the core. The membership degrees less than $\lambda$ are reduced to 0, creating the exclusion region. Finally, all membership degrees between $\lambda$ and $1-\lambda$ are converted to a completely uncertain region spanning the values $[0, 1]$ called the shadow. This balances the uncertainty that was lost by elevation and reduction. Thus, for the SS $\tilde{A}$ the functional mapping of the input space is a three valued logic:

$$\tilde{A}: X \rightarrow \{1, 0, [0,1]\}$$  \hspace{1cm} (5.2)

In order to obtain the optimal $\lambda$ value that balances the uncertainty lost during elevation and reduction, the following simple optimization can be performed:

$$V(\lambda) = \left| \int_{x_{Excl}} \mu(x) \, dx + \int_{x_{Core}} (1-\mu(x)) \, dx - \int_{x_{Sh}} \, dx \right|$$  \hspace{1cm} (5.3)

where, Excl, Core, and Sh denote the exclusion, core and shadow regions of the SS in the input domain. Due to the nature of the optimization function, $\lambda \in [0,0.5]$ and that $V(\lambda_{opt})=0$ where $\lambda_{opt}$ is the optimal value for $\lambda$ [121], [123]. Furthermore, in practical applications where the input domain is usually discretized, the solution is:

$$\lambda_{opt} = \arg\min_{\lambda} V(\lambda),$$  \hspace{1cm} (5.4)

The generated SS $\tilde{A}$ for the initial T1 FS $A$ is shown in Figure 13. The figure also depicts the three regions core, exclusion and shadow of the SS in relation to the T1 FS.
5.2.1.2 Shadowed Type-2 Fuzzy Sets (ST2 FSs)

Similar to SS, a ST2 FS is directly induced by a GT2 FS by transforming all the T1 fuzzy secondary membership functions into their SS forms [121]. Here, all secondary membership functions of the respective GT2 FSs are assumed to be convex fuzzy sets. Hence, the secondary membership functions $f_s(u)$ can be described as:

$$f_s(u) = \begin{cases} 
  g_s(u) & u \in [s_L(x|0), s_L(x|1)], \quad g_s(u) \in [0, 1] \\
  1 & u \in [s_L(x|1), s_R(x|1)] \\
  h_s(u) & u \in [s_R(x|1), s_R(x|0)], \quad h_s(u) \in [0, 1] \\
  0 & \text{Otherwise}
\end{cases}$$

(5.5)

where $u \in J_x$ and $J_x$ is the support of the secondary membership function at $x$. The values $s_L$ and $s_R$ represent the left and right boundaries of the secondary membership function respectively. $g_s(u)$ and $h_s(u)$ are monotonically non-decreasing and monotonically non-increasing functions in their respective domains (see Figure 14 (a)). This assumption of convex secondary membership functions does not severely limit the applicability of ST2 FSs in real world problems as the most commonly used secondary membership functions are all convex [121].

A ST2 FS $\tilde{A}$ is induced by a GT2 FS $\tilde{A}$. The process of constructing $\tilde{A}$ constraints all the secondary membership functions of $\tilde{A}$ to be SSs. The ST2 FS $\tilde{A}$ can be seen as functional mapping:

$$\tilde{A}: X \times [0, 1] \rightarrow \{0, 1, [0, 1]\}$$

(5.6)
Here, the secondary membership of 1 corresponds to the core of the ST2 FSs, secondary membership of 0 corresponds to the exclusion region and the absolutely uncertain grade of $[0, 1]$ corresponds to the shadow of the secondary membership function. The ST2 FS membership function can be expressed as follows:

Figure 14. Generating ST2 FS (a) Secondary membership function of a GT2 FS and its segmentation using two $\alpha$-planes (b) the optimization function $V(\tilde{\lambda})$

Here, the secondary membership of 1 corresponds to the core of the ST2 FSs, secondary membership of 0 corresponds to the exclusion region and the absolutely uncertain grade of $[0, 1]$ corresponds to the shadow of the secondary membership function. The ST2 FS membership function can be expressed as follows:
\[
\tilde{A} = \{(x,u), \mu_{\tilde{A}}(x,u) \mid x \in X, u \in [0,1], \mu_{\tilde{A}}(x,u) \in \{0,1,0\}\}
\] (5.7)

As with constructing SS, the process of constructing an ST2 FS \( \tilde{A} \) based on a GT2 FS \( \tilde{A} \) includes elevation, reduction and balancing of the membership grades. The ST2 FS \( \tilde{A} \) is constructed using a suitable threshold \( \tilde{\lambda} \). The core \( \text{core}(\tilde{A}) \) of ST2 FS \( \tilde{A} \) can be described as a footprint of \( \tilde{A} \) where all secondary membership degrees are greater than \( 1 - \tilde{\lambda} \):

\[
\text{core}(\tilde{A}) = \{(x,u), x \in X, u \in [0,1], \mu_{\tilde{A}}(x,u) > (1 - \tilde{\lambda})\}
\] (5.8)

The exclusion region \( \text{excl}(\tilde{A}) \) of ST2 FS \( \tilde{A} \) can be defined as a footprint of \( \tilde{A} \) where all secondary memberships are less than threshold \( \tilde{\lambda} \):

\[
\text{excl}(\tilde{A}) = \{(x,u), x \in X, u \in [0,1], \mu_{\tilde{A}}(x,u) < \tilde{\lambda}\}
\] (5.9)

Finally, the shadow region \( \text{sh}(\tilde{A}) \) of ST2 FS \( \tilde{A} \) can be constructed as a footprint of \( \tilde{A} \) where all secondary memberships are between thresholds values \( \tilde{\lambda} \) and \( 1 - \tilde{\lambda} \):

\[
\text{sh}(\tilde{A}) = \{(x,u), x \in X, u \in [0,1], \tilde{\lambda} \leq \mu_{\tilde{A}}(x,u) \leq (1 - \tilde{\lambda})\}
\] (5.10)

As before, the optimal value of the threshold should be found. The process of locating the optimal value of threshold \( \tilde{\lambda} \) consists of finding a pair of \( \alpha \)-planes at levels \( \tilde{\lambda} \) and \( 1 - \tilde{\lambda} \), which optimize a fitness function \( V(\tilde{\lambda}) \). The objective function is composed of three components, which express the amount of uncertainty in regions that were reduced \( (v^{s}(\tilde{\lambda})) \), elevated \( (v^{e}(\tilde{\lambda})) \) or balanced \( (v^{b}(\tilde{\lambda})) \). Using the notation depicted in Figure 14 (a) the individual components can be expressed as:
\[
V^R(\tilde{\lambda}) = \int_{x \in X} \int_{u \in \Delta} \mu_{\tilde{x}}(x, u) \, du \, dx + \int_{x \in X} \int_{u \in \Delta} \mu_{\tilde{x}}(x, u) \, du \, dx
\]  
(5.11)

\[
V^E(\tilde{\lambda}) = \int_{x \in X} \int_{u \in \Delta} \mu_{\tilde{x}}(x, u) \, du \, dx
\]  
(5.12)

\[
V^B(\tilde{\lambda}) = \int_{x \in X} \int_{u \in \Delta} \mu_{\tilde{x}}(x, u) \, du \, dx + \int_{x \in X} \int_{u \in \Delta} \mu_{\tilde{x}}(x, u) \, du \, dx
\]  
(5.13)

By combining all three components the optimization function \( V(\tilde{\lambda}) \) can be constructed as:

\[
V(\tilde{\lambda}) = \left| V^R(\tilde{\lambda}) + V^E(\tilde{\lambda}) - V^B(\tilde{\lambda}) \right|
\]  
(5.14)

As in the case of T1, in practical cases where the GT2 FS \( \tilde{A} \) is represented in the \( \alpha \)-plane framework with a finite number of \( \alpha \)-planes, the solution can be obtained as:

\[
\tilde{\lambda}_{opt} = \arg \min_{\tilde{\lambda}} V(\tilde{\lambda})
\]  
(5.15)

An example of the optimization function \( V(\tilde{\lambda}) \) is depicted in Figure 14 (b).

A ST2 FS \( \tilde{A} \) can be completely described using its inner and outer boundaries \( \tilde{A}_i \) and \( \tilde{A}_o \). Each boundary is composed of two T1 fuzzy membership functions, the lower (\( \mu_{\tilde{A}_i}(x) \), \( \mu_{\tilde{A}_o}(x) \)) and the upper (\( \mu_{\tilde{A}_i}(x) \), \( \mu_{\tilde{A}_o}(x) \)) membership functions. The outer boundary marks the boundary between the exclusion and the shadow region. Similarly, the inner boundary marks the transition from the shadow to the core region. This is depicted in Figure 15, which shows a GT2 FS \( \tilde{A} \) and its derived ST2 FS \( \tilde{A} \). This simplified view offers a convenient way to fully describe the ST2 FS \( \tilde{A} \) as:
where both $\tilde{A}_I$ and $\tilde{A}_O$ are IT2 FSs.

5.2.1.3 Set Theoretic Operations of Shadowed Type-2 Fuzzy Sets (ST2 FSs)

In this section, the three elementary operations of intersection, union and complement on ST2 FSs that were presented in [121] are reviewed. Since a ST2 FS can be represented using 2 IT2 FSs, all ST2 set theoretic operations can be reduced to a collection of set theoretic operations on IT2 FSs.

The intersection (meet) of two ST2 FSs $\tilde{A}$ and $\tilde{B}$ can be defined as follows:

$$\tilde{A} \sqcap \tilde{B} = \{\tilde{A}_I, \tilde{A}_O\} \sqcap \{\tilde{B}_I, \tilde{B}_O\} = \{\tilde{A}_I \sqcap \tilde{B}_I, \tilde{A}_O \sqcap \tilde{B}_O\}$$

Thus, the intersection of two ST2 FS can be represented as the intersections of four IT2 FS. The method for calculating intersection of two IT2 FSs, using the t-norm (minimum or product), described in [6] can be used to calculate individual components.
The union (join) of two ST2 FSs $A$ and $B$ can be defined as follows:

$$\tilde{A} \cup \tilde{B} = \{\tilde{A}_j, \tilde{A}_o\} \cup \{\tilde{B}_j, \tilde{B}_o\} = \{\tilde{A}_j, \tilde{B}_j, \tilde{A}_o, \tilde{B}_o\}$$ (5.18)

As before, the union of two ST2 FS can also be represented as the union of four IT2 FSs. Similarly, the method for finding the union of two IT2 FSs, using the t-conorm (maximum or sum), described in [6] can be used to calculate individual components.

Finally, the complement of a ST2 FSs $\tilde{A}$ can be obtained as follows:

$$\tilde{A} = \{\tilde{A}_j, \tilde{A}_o\} = \{\tilde{A}_j, \tilde{A}_o\} \quad \forall x \in X$$ (5.19)
Again, the complement of a ST2 FS can be represented as the complement of two IT2 FSs, and therefore, the method for finding the complement of an IT2 FS described in [6] can again be used here. An example demonstrating the set theoretic operations on two ST2 FSs $\tilde{A}$ and $\tilde{B}$ are depicted in Figure 16.

5.2.1.4 TYPE REDUCTION AND DEFUZZIFICATION OF ST2 FSs

Similar to the basic set theoretic operations, the type-reduction of ST2 FSs also takes advantage of the well-established and computationally efficient algorithms of IT2 FSs. The centroid of an ST2 FS $\tilde{A}$ denoted as $C_{\tilde{A}}$ can be described using two intervals describing the inner and the outer centroids $C_{\tilde{A}}^I$ and $C_{\tilde{A}}^O$:

$$C_{\tilde{A}} = \{C_{\tilde{A}}^I, C_{\tilde{A}}^O\}$$

(5.20)

The inner and the outer centroids can be computed by independently type-reducing the inner and the outer boundary sets $\tilde{A}_I$ and $\tilde{A}_O$. Hence:

$$C_{\tilde{A}} = \{C_{\tilde{A}}^I, C_{\tilde{A}}^O\} = \{c_{I}^i, c_{I}^o, c_{r}^i, c_{r}^o\}$$

(5.21)

The outer centroid $C_{\tilde{A}}^O$ marks the boundary between the exclusion region and the shadowed region of the centroid. Similarly, the inner centroid $C_{\tilde{A}}^I$ creates a boundary between the shadowed boundary and the core region. An example of the centroid of ST2 FS is depicted in Figure 17.
Three different methods for defuzzification of the centroid of ST2 FSs have been proposed in [121]. These are analogous to the defuzzification methods proposed by Pedrycz in [123], and are namely the optimistic ($y^o$), pessimistic ($y^p$) and weighted ($y^w$) defuzzification methods. The output values of each method can be expressed as follows:

$$y^o = \frac{c_i^o + c_j^o}{2}$$  \hspace{1cm} (5.22)

$$y^p = \frac{c_i^o + c_j^o}{2}$$  \hspace{1cm} (5.23)

$$y^w = \frac{\sum_{i=1}^{N} w(x_i) x_i}{\sum_{i=1}^{N} w(x_i)}$$  \hspace{1cm} (5.24)

where, $w(x_i)$ is a specific weighting function, e.g. a trapezoidal weighting function (as depicted in Figure 17).
This section first describes the architecture and inference of the presented ST2 FLSs. Next, the design of the ST2 FLS based on a GT2 FLS is outlined.

An FLS is a rule based system with individual rule antecedents and consequents represented as FSs. While many different types of FLS can be found in literature [16], [29], [30], [31] (e.g. Mamdani or Takagi-Sugeno), this section focuses on the Mamdani type of FLS.
However, it has to be noted that the presented inference method can easily be extended to the Takagi-Sugeno type FLS as well. The architecture of the Mamdani FLS can be decomposed into four major parts: input fuzzification, fuzzy inference engine, fuzzy rule base and output defuzzification or processing, as shown in Figure 18 (a). For the case of the proposed ST2 FLS the fuzzy rule base is populated with linguistic implicative fuzzy rules in the following form:

\[
\text{Rule } R_k: \quad \text{IF } x_1 \text{ is } A^{k}_{1} \text{ AND } \ldots \text{ AND } x_n \text{ is } A^{k}_{n} \text{ THEN } y \text{ is } B^{k}
\]  

(5.25)

Each input \( x_i \) is first fuzzified by computing the membership to the respective antecedent ST2 FS. The membership grade of input \( x_i \) with respect to ST2 FSs \( A^{k}_{i} \) is equal to the secondary membership function of \( A^{k}_{i} \) at coordinate \( x_i \), which can be denoted as \( \mu_{A^{k}_{i}}(x_i) \).

The output of rule \( R_k \) can be expressed as a ST2 fuzzy membership function \( \mu_{R_k}(\bar{x}, y) \), which can be computed by applying the fuzzy meet operations to the rule antecedents and the consequents:

\[
\mu_{R_k}(\bar{x}, y) = \mu_{A_{1}^{k}}(x_1) \prod \ldots \prod \mu_{A_{n}^{k}}(x_n) \prod B^{k}(y)
\]  

(5.26)

The ST2 FSs meet operation \( \prod \) can be computed according to the description provided in Equation (5.17). For completeness sake, Equation (5.26) can be simplified into:

\[
\mu_{R_k}(\bar{x}, y) = [\prod_{i=1}^{n} \mu_{A_{i}^{k}}(x_i)] \prod B^{k}(y)
\]  

(5.27)

Assuming that there are \( K \) distinct fuzzy rules, the output ST2 FSs \( B(y) \) can be computed by aggregating the individual rule outputs via the join operation:
\[ \widetilde{B}(y) = \bigcup_{i=1}^{K} \mu_{\eta_i}(\bar{x}, y) \]  

(5.28)

The ST2 FSs join operation \( \bigcup \) can be computed according to the description provided in Equation (5.18).

Finally, the output ST2 FS \( \widetilde{B}(y) \) is first type-reduced in the first phase of output processing and subsequently defuzzified into a terminal real-valued output in the second phase. As previously discussed, by applying IT2 FSs type-reduction algorithms such as the Enhanced Karnik-Mendel algorithm [46], [39] individually to the inner and the outer boundaries of the output ST2 FSs \( \widetilde{B}(y) \) its centroid \( C_{\widetilde{B}} \) can be obtained. This centroid can then be defuzzified using any of the previously discussed defuzzification techniques for ST2 FSs from Section 5.2.1.4.

By following the described operations of meet, join and type-reduction on ST2 FSs, the entire fuzzy inference process with ST2 FLS can be thought of as a parallel processing of two IT2 FLSs, one for the inner and one for the outer boundary IT2 FSs of the ST2 FSs. The two results are then merged during the defuzzification stage. This interpretation is depicted in Figure 18 (b).

When compared to IT2 FLS, one of the major advantages of the proposed ST2 FLS is that its design is directly induced from a GT2 FLS. As a consequence of the design process the ST2 FLS preserves the uncertainty modeled by the original GT2 FLS. The design process of obtaining an ST2 FLS based on a GT2 FLS can be described in several steps as follows:
Step 1: For all antecedent and consequent GT2 FSs find the optimal splitting $\alpha$-planes $\tilde{\lambda}$ by minimizing function $V(\tilde{\lambda})$ according to Equation (5.14).

Step 2: Convert all antecedent and consequent GT2 FSs into ST2 FSs using the identified splitting $\alpha$-planes $\tilde{\lambda}$.

Step 3: Preserve the fuzzy rule base of the GT2 FLS.
**Step 4:** Modify the fuzzy inference process to implement the fuzzy inference operation on ST2 FSs as described in Equations (5.26) and (5.28).

**Step 5:** Modify the output processing to implement the type-reduction and defuzzification operation on ST2 FSs as described in Section 5.2.1.4.

### 5.3 Experimental Results

This section demonstrates the generation and fuzzy inference process of the presented ST2 FLS on a simple use case. For this purpose a FLS was constructed that mimic a possible real-world fuzzy system. For simplicity, the constructed FLS was composed of two inputs and a single output, all partitioned using three antecedent and consequent FSs, respectively.

First, a GT2 FLS was created which will act as the baseline for the ST2-FLS. The antecedent and consequent GT2 FSs were constructed using triangular primary membership functions with symmetrically positioned Gaussian secondary membership function as shown in Figure 19(a) and Figure 19(b). For the inference process the GT2-FSs shown were decomposed into 200 $\alpha$-planes. The fuzzy rule base used for this experiment is depicted in Table 8.

Second, the GT2 FLS was transformed into a ST2 FLS using the method outlined in section 5.2.2. The optimal values of $\tilde{\lambda}$ values were calculated using equation (5.14) and used to identify splitting $\alpha$-planes of each GT2 FS (Step 2), thus generating the ST2 FSs. The ST2 FSs that were directly induced by the GT2 FSs are shown Figure 20(a) and Figure 20(a), respectively.
Finally, using the inference and defuzzification process detailed in section 5.2.2, the output surface for the ST2 FLS was constructed. The ST2 FLS output surface for the above FLS, computed using the weighted defuzzification method, is shown in Figure 21(a).
comparison, the output surface of the original GT2 FLS is depicted in Figure 21(b). It can be observed that both ST2 and GT2 control surfaces have a similar overall shape.

To further investigate the uncertainty modeling capability of the ST2 FLS it was compared to an IT2 FLS. This IT2 FLS was constructed based on the GT2 FLS outlined above. The individual IT2 FSs were generated using the Footprint-Of-Uncertainty of the GT2 FSs, which is the most common method of generating IT2 FSs using GT2 FSs. The generated IT2 FSs are shown in Figure 22. The rule base shown in Table 8 was used for the IT2 FLS as well. The IT2 FLS output surface and ST2 FLS output surface were then compared to the original GT2 FLS output surface. This comparison was performed by calculating the squared error between the output of the GT2 LFS and the output of either the IT2 FLS or the ST2 FLS, for a given input. In addition, the average computational time of the fuzzy inference process for a single input-output pair achieved by each method was also measured. Table 9 shows the mean

Figure 22. IT2 FSs generated using the FOU of the GT2 FSs, (a) inputs, (b) outputs
Table 9. Comparison of the uncertainty modeling and run-time of ST2 FLS and IT2 FLS with GT2 FLS

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Squared Error</th>
<th>Average Runtime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT2 FLS</td>
<td>-</td>
<td>1.98 ×10^{-1}</td>
</tr>
<tr>
<td>IT2 FLS</td>
<td>3.23 ×10^{-4}</td>
<td>2.27 ×10^{-3}</td>
</tr>
<tr>
<td>ST2 FLS (Optimistic)</td>
<td>1.70 ×10^{-5}</td>
<td>2.34 ×10^{-3}</td>
</tr>
<tr>
<td>ST2 FLS (Pessimistic)</td>
<td>1.12 ×10^{-5}</td>
<td>2.34 ×10^{-3}</td>
</tr>
<tr>
<td>ST2 FLS (Weighted)</td>
<td>3.10 ×10^{-7}</td>
<td>2.34 ×10^{-3}</td>
</tr>
</tbody>
</table>

Figure 23. Squared errors of the IT2 FLS (a) and the ST2 FLS (b) compared to the GT2 FLS
(Note the different scale in the vertical axis)

squared error (MSE), and the computation time of each method. It can be observed that the ST2 FLS models the GT2 FLS output better than IT2 FLS. Furthermore, the computational time of the ST2 FLS significantly lower than the GT2 FLS and is much closer to IT2 FLS computation time. Figure 23(a) and Figure 23(b) show the squared error for each input value for IT2 FLS and ST2 FLS (weighted defuzzification), respectively, which further illustrates
the uncertainty modeling capability of the presented ST2 FLS (note the different scales used in Figure 23). Thus, the results show that the presented ST2 FLS is capable of modeling the GT2 FLS better than IT2 FLS while maintaining the computational efficiency of IT2 FLS.

5.4 Conclusions

This chapter presented the novel concept of Shadowed Type-2 Fuzzy Logic Systems (ST2 FLSs) for better representation and modeling of epistemic uncertainties. The presented ST2 FLS is an FLS that utilizes ST2 FSs to model the inputs and outputs of the FLS. Since the ST2 FSs are directly induced by GT2 FSs, the entire design of the ST2 FLS can be automatically derived from a specific GT2 FLS. The methodology for deriving ST2 FLS using a given GT2 FLS was also defined in this chapter. The primary notion of ST2 FLSs is that they offer improved modeling of uncertainty when compared to IT2 FLSs while also providing efficient computational framework. The experimental results demonstrated the feasibility of the presented concept of ST2 FLS as well as its capabilities. The experimental results showed the automatic generation of ST2 FLS using a GT2 FLS, and its similarities to the initial GT2 FLS. Furthermore, comparison to both IT2 FLS and GT2 FLS in terms of both modeling accuracy and computational complexity demonstrated the superior uncertainty modeling capability of the presented ST2 FLS when compared to IT2 FLS as well as the computational advantage when compared to GT2 FLS. As future work the advantages of ST2 FLS compared to IT2 FLS as well as compromises compared to GT2 FLS can be further investigated by applying the presented methodology to real-world problems.
CHAPTER 6

APPLICATIONS OF FLS FOR UNDERSTANDABILITY AND UNCERTAINTY MODELING

This chapter applies the algorithms and methodologies presented in this dissertation to real-world data and systems. The experimental results presented in this chapter will exemplify the need of the presented algorithms as well as their capability of improving understandability and uncertainty modeling in real-world scenarios.

This chapter first presents algorithms and methodologies for improving understandability applied to real-world datasets. The Fuzzy MF generation and linguistic summarization techniques discussed in Chapter 3 and Chapter 4 were applied to multiple datasets for improving understandability of data. Furthermore, the above techniques were also used to provide information to system users for improved state awareness via understandability.

Second, the novel ST2 FLS presented in Chapter 5 for improved uncertainty modeling is applied to a real-world control scenario. Experiments were carried out that compares the presented method with IT2 FLS and GT2 FLS in control scenarios where noisy data may be present. The improved uncertainty modeling when dealing with noisy input data of the presented ST2 FLS was exemplified in this experiment. Furthermore, the run-time advantage when compared with GT2 FLS was also empirically verified in this experiment.
6.1 Improving Understandability of Real-World Systems

This section contains specific experimental results on improving understandability of data and systems with an emphasis on a real-world usage. The system utilized for demonstrating the understandability improvement of the presented methodologies is a Building Energy Management System (BEMS).

6.1.1 System Description

Buildings consume more than 20% of world energy production and around 40% of US energy production [150], [151]. Such energy consumption means buildings are one of the major causes of greenhouse gas production as well [152]-[155]. Due to various reasons, the energy usage in buildings has been steadily growing [151]. And this number has been projected to further increase [150], [156]. Building Energy Management Systems (BEMSs) are responsible for monitoring building state and controlling HVAC, lighting and other systems within the building. BEMSs are highly complex information gathering and control systems and implement advanced control strategies to improve energy efficiency while maintaining occupant comfort [17], [157]. BEMSs enable significant energy savings in buildings when properly tuned and controlled [158]-[160].

Modern BEMS are extremely complex multi-input multi-output systems that consist of thousands of inter-dependent components such as sensors, controllers and actuators [161]. BEMS uses a large array of sensors installed within the building, outside the building and throughout the air handling systems to gather information about zone temperature, air quality, occupancy, and even lighting [161]-[163]. BEMS uses this information to control the heating,
cooling and lighting of the building [159], [164], [165]. BEMSs are also responsible for provide information about the current state of the system to building managers, who should maintain uninterrupted operation of the HVAC and lighting systems without compromising the occupant comfort or impacting temperature-sensitive equipment. Thus, the information provided by the BEMS should allow the building managers to gain an understanding of the current state of the building operation and to quickly focus on inefficiencies and anomalous sub-optimal behavior [166].

However, due to the complexity and the overwhelming amount of acquired data by the BEMS it is difficult to identify important and sub-optimal building behavior and resolve them accordingly [3], [167], [168]. Significant impact of uncertain factors such as weather and occupancy on building state also make it difficult to identify and predict such behavior using traditional methods [169], [170]. The large number of sensors and the interdependency of measurements make it difficult to identify and locate abnormal behavior or malfunctions [3]. Therefore, inspection of reported data and identification of anomalous behavior and inefficiencies is a daunting task for building managers.

It has been shown in previous work that advanced information representation and visualization techniques can lead to significant savings in energy and identification of hardware faults within buildings [3], [160], [171]-[173]. Furthermore, gathering and analyzing sensor data allows the identification of previously unknown building performance characteristics that may lead to a deeper understanding of the overall building behavior [158]. Therefore, in order to improve the understandability of the BEMS data and to enhance the state-awareness of building managers, the methodologies presented in this dissertation were applied to the BEMS data.
6.1.2 Data Description

Typical BEMS provides measurements from multiple sensors throughout the building. Some measurements are associated with the entire building (e.g. outside air temperature), some are associated with individual floors (e.g. return air temperature or supply air fan load for an air handling unit at a given floor) and some are associated with individual occupant zones on the floor (e.g. zone temperature). Thus, the information collected from the BEMS is arranged in a hierarchical manner, with the building level data at the highest level and are common to all floors and zones. Similarly, the zone level is the lowest level with the highest granularity.

The sensor measurements collected over time constitute a time-series data describing the behavior of each occupant zone. Different patterns of zone behaviors can be experienced in a typical building. For example, a common pattern for winter climates exhibits pre-heating of the building in the morning, regulating appropriate human comfortable temperatures during a day [174], and reducing the set point to maintain lower temperatures at night.

The behavior of each building zone can be described as a feature vector extracted from the sensor measurements. This feature $X(t)$ extracted at time $t$ can then be expressed as:

$$X(t) = \{x_1(t), x_2(t), \ldots, x_N(t)\}$$

(6.1)

where, $x_i(t)$ denotes the specific value of the $i^{th}$ attribute sampled at time $t$ (e.g. zone temperature) and $N$ denotes the dimensionality of the feature vector, i.e. the number of sensors the data is read from.
Table 10. List of extracted attributed and their scope

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Measurement unit</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>HH:MM</td>
<td>Building</td>
</tr>
<tr>
<td>Outside air temperature</td>
<td>°F</td>
<td>Building</td>
</tr>
<tr>
<td>Chiller temperature</td>
<td>°F</td>
<td>Floor</td>
</tr>
<tr>
<td>Mixed air temperature</td>
<td>°F</td>
<td>Floor</td>
</tr>
<tr>
<td>Return air temperature</td>
<td>°F</td>
<td>Floor</td>
</tr>
<tr>
<td>Damper position</td>
<td>%</td>
<td>Floor</td>
</tr>
<tr>
<td>Exhaust fan load</td>
<td>%</td>
<td>Floor</td>
</tr>
<tr>
<td>Exhaust fan current</td>
<td>A</td>
<td>Floor</td>
</tr>
<tr>
<td>Supply fan load</td>
<td>%</td>
<td>Floor</td>
</tr>
<tr>
<td>Supply fan current</td>
<td>A</td>
<td>Floor</td>
</tr>
<tr>
<td>CO₂ level</td>
<td>PPM</td>
<td>Zone</td>
</tr>
<tr>
<td>Zone temperature</td>
<td>°F</td>
<td>Zone</td>
</tr>
</tbody>
</table>

Figure 24. Typical operational characteristics of a selected office room for a 48 hour period
The real-world BEMS was data recorded from an office building in the Pacific Northwest part of the U.S. The building consists of 11 floors, where each floor has between 10 and 60 different measured thermal occupant zones. Various sensors are available throughout the building measuring attributes related to individual thermal zones, entire floors or the entire building.

For the purpose of experimental demonstration, 12 attributes were identified. These attributes together with their scope are listed in Table 10. The data is collected by the system at 45 minute intervals. Data was collected for the month of September from 2 floors of the building. The typical operational characteristics of the temperature sensors shown in Table 10 for a 48 hour period is depicted in Figure 24 for reference.

6.1.3 Anomaly Detection and Graphical User Interface

In addition to applying the techniques described in this dissertation, in order to further increase the state awareness of the building manager, an anomaly detection mechanism, and an easy to read, holistic Graphical User Interface (GUI) was also developed. This section will briefly describe the developed anomaly detection mechanism and the GUI.

The anomaly detection mechanism was developed so that anomalous and potential sub-optimal behavior can be easily identified and reported to the building manager. The GUI makes reporting such anomalous behavior in an understandable format possible.
The anomaly detection mechanism previously presented in [175] and [17] utilizes a simple modified nearest neighbor clustering (NNC) which is a computationally efficient one-pass algorithm for unsupervised modeling of the input data [17]. One of the major advantages of the proposed algorithm is that it is capable of online learning, which means that the model can be updated without the need to re-learn the entire training data set. Furthermore, once the clusters are generated, the complete dataset can be represented using cluster prototypes and new data points only need to be compared against the cluster prototypes, which is suitable for large data sets.

Given a training dataset with normal building behavior, and a maximum cluster diameter, the NNC will generate a set of clusters that represent normal behavior of the building. A previously unseen data point is determined to be an anomaly if it doesn’t fall within any of the clusters generated as the normal behavior. Furthermore, the distance from to the new data point from the closest cluster center of gravity can be used as an indicator of the level of abnormality.

Figure 25. Implemented GUI with (a) building view, (b) floor view, and (c) data view
The implemented GUI is depicted in Figure 25. The GUI contains three main information views: the building view (Figure 25 (a)), the floor view (Figure 25 (b)) and the data view (Figure 25 (c)). The building view provides a summary view of all floors in the building, where color can be assigned to depict various information, such as average floor temperature or the maximum anomaly level. In this figure, the floor view shows the floor plan of the selected floor, where the color of each zone depicts either the average temperature or the confidence that an anomalous behavior was identified for a given zone (using the distance to the closest cluster center). Figure 26 depicts the floor view in different operating conditions.

Finally, the user can select a specific zone for the given floor and observe the source data plotted over time. The building manager can plot multiple sources of data in the data view. Upon selecting a specific building zone, the GUI also linguistically expresses whether the particular zone behaves according to the normal behavior model or whether an anomaly has been identified. If an anomaly has been identified, the linguistic description of the identified anomaly is provided, where the complexity of the generated summaries can be interactively adjusted.

Figure 26. Floor view depicting (a) normal behavior (b) abnormal behavior using temperature as color, and (c) abnormal behavior utilizing anomaly level as color
6.1.4 Experimental Results

The algorithms and techniques described in Chapter 3 and Chapter 4 of this dissertation were applied to the system described above for improving understandability. Fuzzy sets that describe each attribute was generated using the methodology described in Chapter 3. Because the uniqueness of the time attribute, the fuzzy sets depicted in Figure 27, where the attribute is represented using six fuzzy sets with five labels was utilized.

Using the linguistic summarization technique described in Chapter 4, Yager type summaries were generated for the dataset. The top 10 Yager type summaries based on the quality measures is shown in Table 11. It can be observed that the generated summaries are intuitively what can be expected of a BEMS dataset, for example summary 1 and summary 2 suggest that low temperatures are encountered at morning and night times. Furthermore, using the summaries, interdependencies between dimensions may also be discovered; for example, summaries 5 and 6 suggest the exhaust fan load and exhaust fan current may be correlated. In addition to these, performance characteristics of the building may also be derived from the summaries. Summaries 3 and 4 suggest in the afternoon, the building has very high CO\textsubscript{2} levels as well as high zone temperatures which may be due to the occupancy of the building during the day. However, these high levels of CO\textsubscript{2} and temperatures, may cause discomfort to the
occupants towards the end of the work day. Using the generated Yager type summaries, it is also possible to parameterize normal building operation and identify anomalous behavior by generating daily summaries and comparing them to the overall building performance summaries.

The use of linguistic summarization in real-world scenarios were further exemplified by means of providing linguistic summaries for detected anomalous system behavior. The anomaly detection mechanism described in section 6.1.3 was utilized to detect anomalies in the building data. Once an anomaly is detected, an IF-THEN type summary is generated that describes the anomaly. The IF-THEN type summary will give the building manager a clear description of why the behavior was detected as an anomaly, providing the building manager with probable cause and possible actionable information. Here the IF-THEN summary was modified to describe anomaly level using the linguistic label “with confidence” as the consequent, because the consequent will always be the anomaly level.

<table>
<thead>
<tr>
<th>Summary number</th>
<th>Yager type summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Most Very Low Zone Temperature is at Night</td>
</tr>
<tr>
<td>2</td>
<td>Most Very Low Outside Air Temperature is at Morning</td>
</tr>
<tr>
<td>3</td>
<td>Most High Zone Temperature is at Afternoon</td>
</tr>
<tr>
<td>4</td>
<td>Most Very High CO2 is at Afternoon</td>
</tr>
<tr>
<td>5</td>
<td>Most High Exhaust Fan Current is High Exhaust Fan Load</td>
</tr>
<tr>
<td>6</td>
<td>Most Medium Exhaust Fan Current is Medium Exhaust Fan Load</td>
</tr>
<tr>
<td>7</td>
<td>Most Low Damper Position is Low Exhaust Fan Load</td>
</tr>
<tr>
<td>8</td>
<td>Most High Return Air Temperature is High Zone Temperature</td>
</tr>
<tr>
<td>9</td>
<td>Most Medium Zone Temperature is High CO2</td>
</tr>
<tr>
<td>10</td>
<td>Most High Damper Position is Low Outside Air Temperature</td>
</tr>
</tbody>
</table>
The anomalies for the second two weeks of the dataset were detected and a summary report for the detected anomalies was generated, which are shown in Table 12. Since each zone of the building may have different sub-optimal behavior the location of the anomaly is also important. The linguistic summaries give clear and concise information on why each behavior was selected as an anomaly. For example, summary 1 is an anomalous behavior because the recorded zone temperature was very low when the chiller temperature was high, meaning, the intent of the BEMS was not to cool the zone, but the zone was cool nevertheless. Such information also can lead to the identification of the root cause of the problem as well as possible actions to take to mitigate the anomalous behavior.

Table 12. Generated IF-THEN type summaries for the dataset

<table>
<thead>
<tr>
<th>No</th>
<th>Location</th>
<th>Time</th>
<th>IF-THEN type Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Floor 7, Zone 6</td>
<td>9/16, 3:45am – 6:00am</td>
<td>Zone Temperature is Very Low and Chiller Temperature is High with Confidence Very High</td>
</tr>
<tr>
<td>2</td>
<td>Floor 7, Zone 4</td>
<td>9/16, 3:00pm – 6:00pm</td>
<td>Exhaust Fan Load is High and Time is Afternoon with Confidence Very High</td>
</tr>
<tr>
<td>3</td>
<td>Floor 7, Zone 15</td>
<td>9/16, 6:45am – 7:30am</td>
<td>Zone Temperature is Very Low and Mixed Air Temperature is Low with Confidence Significant</td>
</tr>
<tr>
<td>4</td>
<td>Floor 7, Zone 10</td>
<td>9/26, 11:15pm</td>
<td>Time is Night and Supply Fan Current is Very Low with Confidence Very High</td>
</tr>
<tr>
<td>5</td>
<td>Floor 5, Zone 21</td>
<td>9/27, 9:00am</td>
<td>Exhaust Fan Current is Very Low and Return Air Temperature is Low with Confidence Significant</td>
</tr>
<tr>
<td>6</td>
<td>Floor 5, Zone 20</td>
<td>9/27, 11:15pm</td>
<td>Supply Fan Current is Very Low and Exhaust Fan Current is Low with Confidence Very High</td>
</tr>
<tr>
<td>7</td>
<td>Floor 5, Zone 17</td>
<td>9/28, 9:00am – 9:45am</td>
<td>Damper Position is Medium and Return Air Temperature is Low with Confidence Very High</td>
</tr>
<tr>
<td>8</td>
<td>Floor 5, Zone 9</td>
<td>9/28, 9:45pm</td>
<td>Zone Temperature is Very Low and Exhaust Fan Load is Medium with Confidence Very High</td>
</tr>
<tr>
<td>9</td>
<td>Floor 5, Zone 17</td>
<td>9/30, 1:30am</td>
<td>Mixed Air Temperature is Medium and Damper Position is High with Confidence Significant</td>
</tr>
</tbody>
</table>
To evaluate the performance of the linguistic summarization with real-time data, the methodology was tested against several known simulated sub-optimal anomalous behavior. Six different abnormal scenarios were tested and the time each method identified the anomalous behavior is shown in Table 13.

Table 13. Tested anomalous behavior scenarios

<table>
<thead>
<tr>
<th>Case</th>
<th>Type</th>
<th>Fault</th>
<th>Start Time</th>
<th>End Time</th>
<th>Duration (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>Sensor Fault</td>
<td>Constant default sensor value</td>
<td>09/16 22:30</td>
<td>09/17 10:30</td>
<td>12</td>
</tr>
<tr>
<td>Case 2</td>
<td>Constant previous sensor value</td>
<td>09/01 20:00</td>
<td>09/01 08:00</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Case 3</td>
<td>Constant degradation of sensor value</td>
<td>09/16 21:00</td>
<td>09/17 12:00</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Case 4</td>
<td>Physical Abnormality</td>
<td>Open window</td>
<td>09/02 21:00</td>
<td>09/03 09:00</td>
<td>12</td>
</tr>
<tr>
<td>Case 5</td>
<td>External heat source</td>
<td>09/18 21:00</td>
<td>09/19 09:00</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Case 6</td>
<td>Closed air supply vent</td>
<td>09/19 09:00</td>
<td>09/19 21:00</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

Table 14. Generated IF-THEN type summaries for each case and the detected times

<table>
<thead>
<tr>
<th>Case</th>
<th>Linguistic Description</th>
<th>Time Detected</th>
<th>Time Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>Zone Temperature is Very Low</td>
<td>09/16 23:15</td>
<td>09/16 23:15</td>
</tr>
<tr>
<td>Case 2</td>
<td>Return Air Temperature is Medium Mixed Air Temperature is Low Zone Temperature is Very High</td>
<td>09/01 23:15</td>
<td>Not Detected</td>
</tr>
<tr>
<td>Case 3</td>
<td>Return Air Temperature is High Zone Temperature is Very Low</td>
<td>09/17 04:30</td>
<td>09/17 10:15</td>
</tr>
<tr>
<td>Case 4</td>
<td>Chiller Temperature is High Zone Temperature is Very Low</td>
<td>09/03 03:00</td>
<td>09/03 04:15</td>
</tr>
<tr>
<td>Case 5</td>
<td>Chiller Temperature is Very Low Zone Temperature is Very High</td>
<td>09/19 01:15</td>
<td>Not Detected</td>
</tr>
<tr>
<td>Case 6</td>
<td>Chiller Temperature is Very Low Zone Temperature is Very High</td>
<td>09/19 14:15</td>
<td>Not Detected</td>
</tr>
</tbody>
</table>

To evaluate the performance of the linguistic summarization with real-time data, the methodology was tested against several known simulated sub-optimal anomalous behavior. Six different abnormal scenarios were tested and the time each method identified the anomalous behavior is shown in Table 13.
anomalous behavior was recorded for comparison. Furthermore, the detection time of the anomalous behavior of the anomaly detection based linguistic summarization method was compared to a typical threshold based alarm.

The six cases, described in Table 13, were divided into sensor faults and physical abnormalities. The sensor faults (Cases 1, 2 and 3) were simulated by injecting artificial values to the system via the installed communication infrastructure. The physical abnormalities were simulated by actual physical changes to the environment (Case 4: by opening a window, Case 5: using a small portable heater, and Case 6: by closing an air supply vent). All six cases were performed in a small enclosed office room during non-occupied hours.

Table 14 shows the time when each of the methods identified the abnormal behavior along with linguistic descriptions generated. Figure 28 shows each test case and the time each method was able to identify the abnormal behavior. Note that the sensor values plotted in each figure are the sensors identified by the linguistic summarization as relevant for the given scenario.

Case 1 (Figure 28a) where the sensor faults to the default value (in this case 0°F) was immediately identified by both linguistic summarization and threshold based methods. The generated linguistic summary correctly identifies the contributing sensor as zone temperature as value as very low.

Case 2 (Figure 28b) was not identified by the threshold based alarm system since the sensor value does not go outside the preset bounds. However, the anomaly detection was able to identify the abnormal behavior by identifying that the return air and mixed air temperatures were lower compared to the zone temperature, which is represented in the linguistic summary.
Figure 28. Abnormal building behavior tested (a) – (c) sensor based anomalies, (d) – (f) physical anomalies
Case 3 (Figure 28c) was identified by both methods, however, the threshold based alarm system only identified the anomaly after the temperature reached the lower threshold set by the system (which was 60°F). The anomaly detection system was able to identify the behavior 5 hours faster. The generated linguistic summary indicates the reason for detection as the return air temperature being much higher compared to the zone temperature.

Case 4 (Figure 28d) where a window was opened during the night was identified by the anomaly detection system because of the discrepancy between the supply air temperature and the zone temperature as indicated by the generated linguistic summary. Again the threshold based alarm system only identified the anomaly after the zone temperature reached the low alarm threshold.

Case 5 (Figure 28e) was identified by the anomaly detection system because of the high zone temperature while the chiller temperature is very low. The alarm based system was unable to identify the anomaly.

Case 6 (Figure 28f), similar to Case 5 was identified by the anomaly detection system due to the difference in the chiller temperature and the zone temperature. Again, the alarm based system failed to identify the anomaly.

In each case, the generated linguistic summary was able to provide clear and concise information to the user regarding the reason why the case was identified as an anomaly. Identifying such anomalous building behavior faster enables building managers to react to the situation more quickly and more effectively. Furthermore, the generated linguistic descriptions for each of the identified anomalous event enable the user to make more informed decisions,
which can lead to energy savings, higher level of comfort for occupants, as well as mitigate equipment failure due to prolonged exposure to abnormal operation conditions.

6.2 Fuzzy Logic for Uncertainty Modeling in Real-World Systems

This section will contain experimental results detailing the use of ST2 FLS, presented in Chapter 5, for improved uncertainty modeling of real-world system. The real-world system selected in this section is a haptic based robot teleoperation scheme that provides dynamic force-feedback for improved obstacle avoidance [176], [177].

6.2.1 Haptics for Robot Teleoperation

Robot teleoperation entails controlling robots and interacting with the environment from a remote site without direct physical contact. The primary advantage of teleoperation is the ability of controlling a robot in a situation where it is unsafe, difficult or inconvenient for a human to be physically present at the location [178], [179]. Furthermore, by using teleoperation extremely precise and accurate control can be achieved when the task of the robot is dynamic and complex such that it is difficult to complete autonomously [178], [180]. Minimizing collisions that may lead to adverse situations as well as equipment damage in these high complexity tasks is difficult. Thus, these tasks require precise movements and have low threshold for deviation. Furthermore, the control should be impervious to noise present in the sensor data. Therefore, minimizing collisions while maintaining high levels of accuracy and task completion times speed is essential [177].
In order to achieve these goals, providing accurate and useable information to the user about the robot position and orientation as well as the working environment, even with the presence of noisy data is critical for successful and effective teleoperation [176], [177]. The most widely used methods of information presentation to the user are visual and auditory means [181], [182]. However, visual and auditory information might not be sufficient in many cases [183], [184].

Difficulty in modeling accurate information and the need for specialized devices makes it difficult to provide information in addition to audio-visual information [176], [177]. However, providing tertiary information via the sense of touch, known as haptics, has gained much interest in recent years [176], [177], [182], [183], [185], [186]. Typically, in haptic applications, a haptic device which is a bi-directional human interface that provides sensory input to the user via the sense of touch while providing control inputs to the machine is used [176], [177], [186], [187]. Two types of haptic devices exist: tactile and kinesthetic [185], [188]. Tactile devices are based on sense of touch and enable the operator to feel textures, friction and rubbing forces, and consistency of objects [183], [185]. Kinesthetic devices reflect forces and increase the state awareness of operators about the work space and potential hazards [185].

Thus haptics has been used in a wide area of robot remote teleoperation tasks [189] such as teleoperation of mobile robots [190] operating industrial robotic manipulators [191], remote non-invasive surgery [192], [193], path planning [194] and virtual sculpting [186], [195]. Furthermore, improvement in time and accuracy of task completion has been shown by utilizing haptics as an additional sensory input to the operator [176], [177], [182], [188], [192], [196].
Virtual force field based force-feedback generation where a force field is modeled surrounding the objects and the manipulator is one of the most commonly used methodologies of force-feedback generation in haptic applications [185], [194]. Other techniques used in literature include accurate physics based models [197], and mass-spring models [198], and PD and PID control methods with varying gains [198]-[201].

6.2.2 System Description

When small precise movements are required from the robot manipulator, the force-feedback should be adjusted for easier maneuverability while maintaining state awareness. Static force field based methods, however, lack the flexibility required when operating in such conditions to adjust the force-feedback. Therefore, the system presented in this section dynamically modifies the force field for providing kinesthetic feedback to operator using a Fuzzy Logic based model [176]. Thus, the presented method is capable of maintaining the state awareness of the operator while improving the accuracy of operation, in sub-optimal noisy scenarios. The presented method utilizes the distance to obstacles and the speed of the robotic manipulator to dynamically vary the force field of the manipulator components to suit the environment and type of movement required for the situation. The presented method was against different levels of noise to identify the uncertainty modeling capabilities of the ST2 FLS compared to GT2 FLS and IT2 FLS.
6.2.2.1 Virtual Force-Field Based Force-Feedback Calculation

Virtual force field has been widely used as a method for generating force-feedback for obstacle avoidance in robot teleoperation applications [176], [185], [194]. Here, the virtual force field of a given object $K$, is represented by a Gaussian function for each axis, centered at $K$. Thus the force field of object $K$ in axis $i$ ($F_{K,i}$) can be expressed as:

$$F_{K,i}(P_i) = \exp \left( -\frac{(K_i - P_i)^2}{2\sigma^2} \right)$$  \hspace{1cm} (6.2)

where $K_i$ is the position of object $K$ in axis $i$ and $P_i$ is a point in axis $i$. The value $\sigma$ expresses the spread of the force field and can be assigned according to prior knowledge about the importance and sensitivity of the object. A force field where $\sigma = 0.2$ is shown in Figure 29 (a).

Since the Gaussian function is asymptotically approaching 0, in order to calculate the distance from an object $J$ to the force field of object $K$ ($J \neq K$), a force threshold, $\delta_F$ is used:

$$T_{K,i} = \{ \min(P_i) | F_{K,i}(P_i) \geq \delta_F, \max(Q_i) | F_{K,i}(Q_i) \geq \delta_F \}$$  \hspace{1cm} (6.3)

where $P_i$ and $Q_i$ are points in axis $i$, and $T_{K,i}$ is the distance threshold for object $K$ in axis $i$ (see Figure 29 (a)).

The force-feedback for each time step $t$ is generated using the force fields of the obstacles and the robot manipulator at time step $t+1$. The force fields of the obstacles are assumed to be static (constant $\sigma$) and represented using the method described above. The force field of each component of the manipulator is dynamically generated at each time step using the method described in section 6.2.2.2 below.
Once the force fields are dynamically generated for all components of the manipulator, the forces acting upon each component of the manipulator for each axis are generated. This is done using the maximum overlap of component and obstacle force fields (Figure 29(b)). The overlap between object $K$ and component $C$ in axis $i$, $C^kM_i$, can be defined as:

$$C^kM_i = \max(F_{K,i}) \mid F_{K,i}(P_i) = F_{C,i}(P_i)$$  \hspace{1cm} (6.4)

Thus, for each component in the manipulator, the overlap of force fields for each obstacle is calculated. For a given axis $i$, the force exerted by an obstacle $K$ on the manipulator component $C$ is thus proportional to the obstacle-component force field overlap in axis $i$.

$$C^k\hat{F}_i \propto C^kM_i$$  \hspace{1cm} (6.5)

where $C^k\hat{F}_i$ is the force exerted by obstacle $K$ on component $C$ in the axis $i$. The forces acting on a given component for all obstacles are calculated and the final force for a component for axis $i$ is the sum of all forces acting upon that component for axis $i$.

$$C^k\vec{F}_i = \sum_{k=1}^{\alpha} C^k\hat{F}_i$$  \hspace{1cm} (6.6)
where, $^{C}F_i$ is the force acting upon component $C$ and $O$ is the number of obstacles in the work space.

Finally the force acting upon each component for a given axis $i$, is aggregated and sent to the force-feedback input device. The forces are aggregated using a weight that is proportional to the velocity of that component, thus exerting more force to the user for components that are moving faster.

$$\tilde{F}_i = \sum_{c=1}^{M} F_{i,c} \cdot w_{c,i}$$

(6.7)

where, $\tilde{F}_i$ is the force sent to the user via the force-feedback device in axis $i$. $M$ is the number of components in the manipulator, and $w_{c,i} \propto V_{c,i}$ where $V_{c,i}$ is the velocity of component $C$ in axis $i$.

6.2.2.2 Fuzzy Based Dynamic Virtual Force-Field Generation

The overall framework of the presented method is detailed in Figure 30. A force-feedback enabled joystick device is used as the input device as well as providing kinesthetic feedback to the user. The position and size of the obstacles in the operating area is assumed to be known. As mentioned above, each obstacle is surrounded by a static virtual force field.

The user inputs the desired location of the manipulator using the force-feedback enabled joystick device. Once a movement of the joystick is made at time $t$, the required actuator angles for the desired movement of the manipulator at time $t + 1$, are calculated. Here,
the actuator angles are calculated using inverse kinematics. Using the generated angles, the position of each component of the manipulator at time $t+1$ is calculated.

Using the calculated position of the manipulator at time $t+1$, and the force fields of obstacles, the closest distance from the manipulator components to the obstacle force fields is calculated, before physically moving the manipulator. This calculation is done for each axis separately. Further, using the position of the manipulator components at time $t$ and time $t+1$, the velocity vector for each component is also calculated. Thus, for each of the components in the manipulator, for each axis, the minimum distance to an obstacle force field, and the velocity is known. These distance and velocity pairs are then passed on to the appropriate fuzzy force field generation system. The fuzzy system then generates the appropriate force field of each component for each axis for time $t+1$. The generated force fields of components for time $t+1$, and the force fields of obstacles are then used to generate the force-feedback for time $t+1$, for each axis using the method described in section 6.2.2.1.
The fuzzy logic based force field generator utilizes the generated distance vector at time $t+1$ and the velocity vector between time $t$ and $t+1$ to dynamically update the force fields of each component. The force field is dynamically updated for components to enable dynamic force-feedback according to the environment and movement speed. This is achieved by weighing the spread of the Gaussian function ($\sigma$ in (6.2)) of manipulator component force fields. At higher speeds of movement, the likelihood of collisions increase. Furthermore, earlier warning is required for the user to react when the actuator is moving faster. Thus the fuzzy system increases the spread of the force field at higher manipulator speeds. When obstacles are further away from the manipulator the spread of the force field can be increased for increased awareness of the environment, without affecting the accuracy.

However, for precise and accurate movements in constricted environments, the larger force field is unsuitable as it yields higher force-feedback that increases the difficulty of precise movements. Thus at lower speeds and with closer obstacles, the spread of the component force fields are reduced, thus enabling movements with higher accuracy, with minimal loss of state awareness.

For axis $x$, $y$ and $z$, a separate force field generator is used, which generates the force field on that axis for the specific component. Each force-feedback generator takes the axis component of the velocity vector and the axis component of the distance vector. The axis component of the distance vector is the distance from the manipulator to the closest obstacle in that axis. By utilizing separate controller for each axis, the dynamic behavior of the force field in that axis can be controlled differently. Differences in fuzzy controllers entail different
rule bases as well as different fuzzy sets that granulizes the input and output spaces differently. For example, a manipulator with limited movement in one axis can have a different force field generator for that axis. Similarly, for manipulator components that behave differently, different force field generators can be used.

Once the force-feedback for each axis is generated it is sent to the user via the force-feedback enabled joystick device as a kinesthetic input. Simultaneously the manipulator is moves to the desired position for time $t + 1$.

6.2.3 Experimental Results

A simple 3-DOF robot manipulator with 3 actuators that was implemented using Lego NXT [202] and utilized in [176], [177] was used for testing. The schematic of the implemented manipulator is depicted in Figure 31(a). Inverse kinematics was used to derive the angles of the actuators for a given end-effector position. The actuator angles can be read via the NXT
interface for calculating the actual position of the robot after a movement has been made. As the force-feedback enabled joystick device, the Novint Falcon device (Figure 31(b)) was selected [203], [204]. The Novint Falcon device has 3 degrees of freedom and 3 actuators work in conjunction to provide powerful and accurate kinesthetic or tactile feedback to the user, while maintaining highly accurate control. As mentioned in section 6.2.2.2, the force field of each axis in the control space is controlled by a separate fuzzy controller. Each controller can be customized according to the freedom of movement in that axis. For the sake of simplicity, the same FLS (fuzzy rules and fuzzy sets) was used for each axis in this section.

A GT2 FLS containing 4 input GT2 FSs for the distance input, 3 input GT2 FSs for the velocity inputs, and 4 output GT2 FSs was selected as the baseline fuzzy system for generating the dynamic force-field and for modeling uncertainty. The fuzzy rule-base used in this section is summarized in Table 15. Figure 32 depicts the input and output GT2 FSs used in the baseline GT2 FLS. Figure 33 depict the IT2 FSs generated from the FOUs of the GT2 FSs, and represent the typical method of approximating GT2 FLSs for real-world usage [19].

Table 15. Fuzzy rule-base used to dynamically generate the force-field in each axis

<table>
<thead>
<tr>
<th>Velocity axis $i$</th>
<th>Distance axis $i$</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low</strong></td>
<td></td>
<td>Medium Low</td>
<td>Medium Low</td>
<td>Medium Low</td>
</tr>
<tr>
<td><strong>Low</strong></td>
<td></td>
<td>Low</td>
<td>Medium Low</td>
<td>Medium Low</td>
</tr>
<tr>
<td><strong>Medium</strong></td>
<td></td>
<td>Medium Low</td>
<td>Medium High</td>
<td>Medium High</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td></td>
<td>Medium High</td>
<td>Medium High</td>
<td>High</td>
</tr>
</tbody>
</table>
Figure 32. GT2 Fuzzy Sets for (a) distance, (b) velocity and (c) force-field spread
Figure 33. IT2 Fuzzy Sets for (a) distance, (b) velocity and (c) force-field spread
Figure 34. ST2 Fuzzy Sets for (a) distance, (b) velocity and (c) force-field spread
Figure 34 depict the ST2 FSs, generated using the method described in Chapter 5, and are directly induced from the original GT2 FSs such that the inherent uncertainty of the GT2 FSs are retained. The IT2 FLS and the ST2 FLS also utilizes the rule-base summarized in Table 15. Two different experiments were carried out to identify the uncertainty modeling capability of the presented ST2 FLS. In the first experiment, different levels of noise was injected to the input variables and the output was calculated using each method for the noisy and noiseless data. The calculated output for the noisy data was then compared to the actual output that should have been generated given the noiseless data in order to determine the uncertainty modeling capability of each method. This process was iterated 5 times over the input space and the resulting error was aggregated over different levels of noise.

Figure 35 depicts the performance of each method at different levels of noise. For this experiment noise levels from Signal-to-Noise-Ratio 20dB – 10dB was introduced to the data (see equation (3.38) in section 3.4.1). As expected, GT2 FLS performs best at high noise levels.
depicting its superior uncertainty modeling. The presented ST2 FLS demonstrates superior uncertainty modeling capability when compared to IT2 FLS as the ST2 FLS yield lower errors at higher noise levels when compared to IT2 FLS.

For the second experiment, a simple task was set up to evaluate the performance of each of the methods given different levels of noise. The selected task comprised of moving the end effector of the manipulator through an area surrounded by an obstacle and is shown in Figure 36. The end effector is entered into the obstacle area starting at point A (Figure 36). At this point the width of the opening is 10 cm. At point C, the width of the opening is reduced to 5 cm simulating a constricted work area. The work area was divided in to three areas: 1) A-B: lower restriction allowing more freedom of movement, 2) B-D: transitioning from a lower restriction area to a higher restriction area, and 3) D-E: high restriction area where more precision of movement is required.

The time to complete each area as well as the complete task was recorded. Furthermore, the accumulated accuracy of the movement of the end effector was calculated as a measure of
distance from the preset furthest distance path shown in Figure 36. The task was completed 5 times using each FLS at different noise levels, and the final results were averaged. Due to time limitations, only 3 levels of noise was selected for this experiment.

Table 16 and Table 17 summarize the results of the second experiment at different noise levels. All three methods perform equally well at little to no noise levels. However, at medium to high noise levels, the presented ST2 FLS outperforms the IT2 FLS in terms of task completion times as well as the averaged accuracy. The task completion time was not significantly affected by low and medium noise levels as the operator was able to maneuver the robot arm through the obstacle without much difficulty even with the presence of noise. It can be observed that in the areas B-D and D-E, the high level of noise affects the accuracy of the task. This can be attributed to the constrained freedom of movement in these areas where the obstacle is closer to the robot arm. The users also noted a noticeable difference in the haptic feedback of the IT2 FLS at higher noise levels, especially during these constrained areas. The standard deviation of the accuracy indicates consistency of the operation. Furthermore, the performance of the ST2 FLS is extremely close to the GT2 FLS which has the highest uncertainty modeling capability.

The average time to compute at output given an input pair was also measured for each method and the GT2 FLS took an average of 0.192 seconds to produce an output (for 200 $\alpha$ -planes). The averaged time for ST2 FLS to produce an output was 0.007 seconds, while the IT2 FLS took only 0.001 seconds. Thus, the presented ST2 FLS is capable of achieving levels of uncertainty modeling that is close to that of GT2 FLS, while maintaining the advantage of faster processing times associated with IT2 FLS.
Table 16. Average task completion times for each method at different noise levels

<table>
<thead>
<tr>
<th>Noise level</th>
<th>Method</th>
<th>A - B (s)</th>
<th>B - D (s)</th>
<th>D - E (s)</th>
<th>Total (mean) (s)</th>
<th>Total (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No noise</td>
<td>IT2 FLS</td>
<td>7.22</td>
<td>2.22</td>
<td>7.61</td>
<td>17.05</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>ST2 FLS</td>
<td>7.25</td>
<td>2.21</td>
<td>7.58</td>
<td>17.04</td>
<td>0.31</td>
</tr>
<tr>
<td></td>
<td>GT2 FLS</td>
<td>7.24</td>
<td>2.23</td>
<td>7.57</td>
<td>17.04</td>
<td>0.29</td>
</tr>
<tr>
<td>20 dB (Low)</td>
<td>IT2 FLS</td>
<td>7.24</td>
<td>2.30</td>
<td>8.85</td>
<td>18.39</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>ST2 FLS</td>
<td>7.23</td>
<td>2.25</td>
<td>8.03</td>
<td>17.51</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>GT2 FLS</td>
<td>7.23</td>
<td>2.26</td>
<td>8.04</td>
<td>17.53</td>
<td>0.41</td>
</tr>
<tr>
<td>12.2 dB (Medium)</td>
<td>IT2 FLS</td>
<td>7.20</td>
<td>2.55</td>
<td>10.07</td>
<td>19.82</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>ST2 FLS</td>
<td>7.24</td>
<td>2.56</td>
<td>8.59</td>
<td>18.39</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>GT2 FLS</td>
<td>7.26</td>
<td>2.39</td>
<td>8.51</td>
<td>18.16</td>
<td>0.42</td>
</tr>
<tr>
<td>10 dB (High)</td>
<td>IT2 FLS</td>
<td>7.31</td>
<td>2.98</td>
<td>12.07</td>
<td>22.36</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td>ST2 FLS</td>
<td>7.28</td>
<td>2.61</td>
<td>8.72</td>
<td>18.61</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>GT2 FLS</td>
<td>7.29</td>
<td>2.58</td>
<td>8.61</td>
<td>18.48</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 17. Average accuracy compared to the furthest distance path for each method at different noise levels

<table>
<thead>
<tr>
<th>Noise level</th>
<th>Method</th>
<th>A - B</th>
<th>B - D</th>
<th>D - E</th>
<th>Total (mean)</th>
<th>Total (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No noise</td>
<td>IT2 FLS</td>
<td>2.06</td>
<td>0.85</td>
<td>0.96</td>
<td>3.87</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>ST2 FLS</td>
<td>2.08</td>
<td>0.84</td>
<td>0.96</td>
<td>3.88</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>GT2 FLS</td>
<td>2.17</td>
<td>0.85</td>
<td>0.93</td>
<td>3.95</td>
<td>0.16</td>
</tr>
<tr>
<td>20 dB (Low)</td>
<td>IT2 FLS</td>
<td>2.08</td>
<td>0.84</td>
<td>1.77</td>
<td>4.69</td>
<td>0.34</td>
</tr>
<tr>
<td></td>
<td>ST2 FLS</td>
<td>2.08</td>
<td>0.85</td>
<td>1.08</td>
<td>4.01</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>GT2 FLS</td>
<td>2.09</td>
<td>0.85</td>
<td>1.03</td>
<td>3.97</td>
<td>0.18</td>
</tr>
<tr>
<td>12.2 dB (Medium)</td>
<td>IT2 FLS</td>
<td>3.12</td>
<td>1.76</td>
<td>2.98</td>
<td>7.86</td>
<td>1.01</td>
</tr>
<tr>
<td></td>
<td>ST2 FLS</td>
<td>2.33</td>
<td>1.35</td>
<td>1.41</td>
<td>5.09</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td>GT2 FLS</td>
<td>2.15</td>
<td>1.24</td>
<td>1.37</td>
<td>4.76</td>
<td>0.39</td>
</tr>
<tr>
<td>10 dB (High)</td>
<td>IT2 FLS</td>
<td>3.59</td>
<td>2.81</td>
<td>3.55</td>
<td>9.95</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td>ST2 FLS</td>
<td>3.21</td>
<td>1.96</td>
<td>2.01</td>
<td>7.18</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td>GT2 FLS</td>
<td>3.11</td>
<td>1.87</td>
<td>1.88</td>
<td>6.86</td>
<td>0.99</td>
</tr>
</tbody>
</table>
6.3 Conclusions

This chapter demonstrated the real-world usability of the understandability and uncertainty modeling methods presented in this dissertation by applying them to real-world data and systems. The improved understandability gained from utilizing the methodologies presented was demonstrated using a building energy management dataset. The ability of the presented methods to generate simple, human understandable linguistic summaries for large complex data sets, as well as the capabilities of the methodologies to provide clear and concise information that improves state-awareness was demonstrated in this chapter. Improved uncertainty modeling of the presented ST2 FLS was demonstrated using a fuzzy force-feedback architecture for obstacle collision avoidance in robot tele-operation. The experimental results demonstrated the improved uncertainty modeling capability of the presented ST2 FLS compared to IT2 FLS as well as the time complexity advantage compared to GT2 FLS.
CHAPTER 7

CONCLUSION AND FUTURE WORK

This chapter provides the final conclusions of the presented work and propose several directions for future work.

7.1 Final Conclusions

The primary motivation for this dissertation was to address and improve the understandability and uncertainty modeling of data using fuzzy systems, through several algorithms and techniques that focus on these aspects as well as improving computational complexity.

A data driven technique for generating fuzzy Membership Functions (MFs) was presented in Chapter 3 that focus on the understandability of the generated MFs. Two new understandability metrics were presented in this chapter and these along with several other understandability metrics were utilized to measure the overall understandability of the generated MFs. The presented method utilizes a classical statistical approach coupled with a deterministic methodology to generate MFs that conform to and optimize MFs based on their understandability. The presented method was tested on several artificial and real-world datasets. Experimental results showed the capability of the presented method to generate MFs that conform to all understandability metrics as well as intuitively reasonable MFs.
Furthermore, MFs generated at different noise levels demonstrated the validity of the generated MFs.

Chapter 4 presented a Self-Organizing Map (SOM) based linguistic summarization method (SOM-LS) method for generating *IF-THEN* and *Yager* type linguistic summaries of data. The presented method utilizes the data compression, approximation and generalization capabilities of SOM to generate noise robust human understandable linguistic summaries of data in an efficient manner. Two new quality measures along with methodology to adopt these and other quality measures to be used with SOM were also introduced in this chapter. An evaluation of the computational complexity of the presented method highlights the potential advantages of the presented method compared to the exhaustive method. The evaluation of the computational complexity also yields a method to identify whether the presented SOM-LS usable for a given dataset and what is the maximum size of the SOM that can guarantee faster summarization of data. SOM-LS method was compared to the exhaustive summarization using artificial and real-world benchmark datasets. The empirical results verified the runtime advantage of the SOM-LS. Furthermore, the noise robust summarization of data using SOM-LS was demonstrated by introducing different levels of noise into the datasets.

Shadowed Type-2 Fuzzy Logic Systems (ST2 FLSs) were presented in Chapter 5 for better representation and modeling of epistemic uncertainties. ST2 FLSs are directly generated from the more capable but computationally expensive GT2 FLSs. Methodology for deriving ST2 FLS from GT2 FLS was presented in Chapter 5 that retains the overall uncertainty modeling capabilities of the original GT2 FLS. The presented ST2 FLS and the methodology for deriving ST2 FLS was experimentally verified using a simple GT2 FLS. Furthermore, the generated ST2 FLS was compared to a IT2 FLS that was generated from the same GT2 FLS.
The experimental results verified the ST2 FLS generation method as well as exemplifying the computational advantage of ST2 FLS when compared to GT2 FLS and the uncertainty modeling capability improvement of ST2 FLS when compared to IT2 FLS.

The relevance and usability of the presented understandability and uncertainty improvement methods in real-world scenarios were presented in Chapter 6. The understandability improvement methods presented in Chapter 3 and Chapter 4 were applied to a real-world building energy management systems dataset. It was demonstrated that by utilizing these methods users can gain an improved understandability of the overall system and its operation. Furthermore, utilizing the methods to summarize potential sub-optimal behavior of the system yields clear and concise information about such behavior and possible actionable information on how to mitigate such behavior. The improved run-times of the presented systems also yield the possibility of real-time summarization of the system state for improved state awareness. The ST2 FLS for improved uncertainty modeling presented in Chapter 5 was applied to a force-feedback based robot teleoperation system. The experimental results showed the superior performance of ST2 FLS in terms of improved and consistent accuracy and task completion times compared to IT2 FLS with epistemic uncertainties in the system. The presented ST2 FLS was also shown to have faster computation times compared to GT2 FLS.

7.2 Future Work

Future work entails applying the presented methods to additional real-world scenarios and further comparison of the methods to other state-of-the-art techniques and algorithms. More specific future directions for each of the methods presented follows.
The presented understandability based MF generation method can be further extended to include other types of MFs such as Gaussian or Bell shaped MFs. Coupled with additional understandability metrics, the presented method can also be extended to generate Type-2 MFs including ST2 FSs. The presented method can be applied to widely used classification and control tasks to investigate the understandability vs performance trade-offs. Furthermore, this type of study can be used to extend the presented method to enable users to determine the trade-off between understandability and performance and generate MFs that suit user requirements.

The SOM-LS linguistic summarization technique can be coupled with other methods presented in the literature for improving the linguistic summarization process, such as heuristic search of the possible summaries, utilizing domain knowledge for constraining the search space etc. The advantage of utilizing the presented SOM-LS method in large continuously growing datasets should be evaluated as the summarizing process is independent from the number of data points in the dataset. The advantages of state-awareness gained by providing real-time system diagnostics via linguistic summaries can also be investigated using such a system.

The advantages of the presented ST2 FLS compared to IT2 FLS needs to be further evaluated by applying the ST2 FLS to real-world systems where uncertainty modeling is required. The computational advantages of the ST2 FLS can be better emphasized in embedded applications. Further investigation of the uses of ST2 FSs such as in linguistic summarization and clustering applications may further solidify the importance of ST2 FSs.
APPENDIX A

LIST OF PUBLICATIONS BY THE AUTHOR

This appendix presents an overview of the author’s published or submitted journal and peer-reviewed conference publications.
Journal Publications


**Abstract:** Building automation systems (BAS), or building control systems (BCS), typically consist of building energy management systems (BEMSs), physical security and access control, fire/life safety, and other systems (elevators, public announcements, and closed-circuit television). BEMSs control heating, ventilation, and air conditioning (HVAC) and lighting systems in buildings; more specifically, they control HVAC’s primary components such as air handling units (AHUs), chillers, and heating elements. BEMSs are essential components of modern buildings, tasked with seemingly contradicting requirements—minimizing energy consumption while maintaining occupants’ comfort. In the United States, about 40% of total energy consumption and 70% of electricity consumption are spent on buildings every year. These numbers are comparable to global statistics that about 30% of total energy consumption and 60% of electricity consumption are spent on buildings. Buildings are an integral part of global cyber physical systems (smart cities) and evolve and interact with their surroundings.


**Abstract:** Resiliency and improved state-awareness of modern critical infrastructures, such as energy production and industrial systems, is becoming increasingly important. As control systems become increasingly complex, the number of inputs and outputs increase. Therefore, in order to maintain sufficient levels of state-awareness, a robust system state monitoring must be implemented that correctly identifies system behavior even when one or more sensors are faulty. Furthermore, as intelligent cyber adversaries become more capable, incorrect values may be fed to the operators. To address these needs, this paper proposes a fuzzy-neural data fusion engine (FN-DFE) for resilient state-awareness of control systems. The designed FN-DFE is composed of a three-layered system consisting of: 1) traditional threshold based alarms; 2) anomalous behavior detector using self-organizing fuzzy logic system; and 3) artificial neural network-based system modeling and prediction. The improved control system state-awareness is achieved via fusing input data from multiple sources and combining them into robust anomaly indicators. In addition, the neural network-based signal predictions are used to augment the resiliency of the system and provide coherent state-awareness despite temporary unavailability of sensory data. The proposed system was integrated and tested with a model of the Idaho National Laboratory's hybrid energy system facility known as HYTEST. Experiment results demonstrate that the proposed FNDFE provides timely plant performance monitoring and anomaly detection capabilities. It was shown that the system is capable of identifying intrusive behavior significantly earlier than conventional threshold-based alarm systems.

**Abstract:** Building Energy Management Systems (BEMSs) are essential components of modern buildings that are responsible for minimizing energy consumption while maintaining occupant comfort. However, since indoor environment is dependent on many uncertain criteria, performance of BEMS can be suboptimal at times. Unfortunately, complexity of BEMSs, large amount of data, and interrelations between data can make identifying these suboptimal behaviors difficult. This paper proposes a novel Fuzzy Anomaly Detection and Linguistic Description (Fuzzy-ADLD)-based method for improving the understandability of BEMS behavior for improved state-awareness. The presented method is composed of two main parts: 1) detection of anomalous BEMS behavior; and 2) linguistic representation of BEMS behavior. The first part utilizes modified nearest neighbor clustering algorithm and fuzzy logic rule extraction technique to build a model of normal BEMS behavior. The second part of the presented method computes the most relevant linguistic description of the identified anomalies. The presented Fuzzy-ADLD method was applied to real-world BEMS system and compared against a traditional alarm-based BEMS. Six different scenarios were tested, and the presented Fuzzy-ADLD method identified anomalous behavior either as fast as or faster (an hour or more) than the alarm based BEMS. Furthermore, the Fuzzy-ADLD method identified cases that were missed by the alarm-based system, thus demonstrating potential for increased state-awareness of abnormal building behavior.


**Abstract:** Artificial Neural Networks (ANN) have been used in the past to predict the performance of printed circuit heat exchangers (PCHE) with satisfactory accuracy. Typically published literature has focused on optimizing ANN using a training dataset to train the network and a testing dataset to evaluate it. Although this may produce outputs that agree with experimental results, there is a risk of over-training or overlearning the network rather than generalizing it, which should be the ultimate goal. An over-trained network is able to produce good results with the training dataset but fails when new datasets with subtle changes are introduced. In this paper we present EBaLM-OTR (error back propagation and Levenberg-Marquardt algorithms for over training resilience) technique, which is based on a previously discussed method of selecting neural network architecture that uses a separate validation set to evaluate different network architectures based on mean square error (MSE), and standard deviation of MSE. The method uses k-fold cross validation. Therefore in order to select the optimal architecture for the problem, the dataset is divided into three parts which are used to train, validate and test each network architecture. Then each architecture is evaluated according to their generalization capability and capability to conform to original data. The method proved to be a comprehensive tool in identifying the weaknesses and advantages of different network architectures. The method also highlighted the fact that the architecture with the lowest training error is not always the most generalized and therefore not the optimal. Using the method the testing error achieved was in the order of magnitude of within 10−5–10−3. It was also show that the absolute error achieved by EBaLM-OTR was an order of magnitude better than the lowest error achieved by EBaLM-THP.
Peer-Reviewed Conference Publications


**Abstract:** Buildings are known to be significant energy consumers throughout the world. Thus, improving the energy efficiency of buildings is a key research goal. However, maintaining occupant comfort while improving energy efficiency in buildings requires close monitoring of the building environment and immediate control actions taken when sub-optimal behavior is identified. Such monitoring requires high frequency data from sensors. Therefore, increasing the data collection rate or the temporal resolution of sensors can lead to improved building control and state-awareness. This paper presents an on-line learning, data-fusion based methodology that uses Artificial Neural Networks (ANNs) to increase temporal resolution of building sensor data. The presented method utilizes sensor information from different sensors in the building to predict higher temporal resolution data of specific sensors. Furthermore, the presented method is capable of learning changing building behavior for improved long-term accuracy. The presented method was applied to a real-world building dataset and was shown to be able to predict high temporal resolution data with a higher accuracy compared to classical methods. Furthermore, the on-line learning was shown to increase the prediction accuracy in long-term operation.


**Abstract:** Heating, Ventilation and Air Conditioning (HVAC) system is largest energy consumer in buildings. Worldwide, buildings consume 20% of the total energy production. Therefore, increasing efficiency of the HVAC system will result in significant financial savings. As one solution, Thermal Energy Storage (TES) tanks are being utilized with buildings to store excess energy to be reused later. An optimal control strategy is crucial for optimal usage. Therefore, this paper presents a novel control framework based on Artificial Neural Networks (ANN) for optimally controlling a TES for achieving increased savings. The presented ANN controller utilizes 3 main inputs: 1) current TES energy availability, 2) predicted building power requirement, and 3) predicted utility load/price. In addition to the design details of the control framework, this paper presents implementation details of the ANN controller. Further, experiments on several test cases were carried out and the paper presents the experimental setup and obtained results for each test case. Performance of the presented ANN control framework was compared against a classical proportional derivative (PD) controller. It was observed that the presented framework resulted in better cost savings than the classical controller consistently for all the experimental test cases.

**Abstract:** Dependency of the transport sector on fossil fuels is encouraging a significant amount of research into improving fuel efficiency in vehicles. Three primary techniques are identified for vehicle fuel efficiency improvement: 1) vehicle technology improvements such as drivetrain improvements, 2) traffic infrastructure improvements such as traffic flow management and route selection, and 3) driver behavior changes such as acceleration and deceleration profiles. Out of the 3 techniques, driver behavior changing has the least implementation cost and is able to provide immediate results. Thus, this paper presents a fuel efficient driving behavior identification and feedback architecture that is specific to fleet vehicles. The presented method utilizes historical data from fleet drivers on specific routes and generates fuel optimal velocity profiles that do not affect travel time. The identified velocity profile is the prompted to the driver via a low-cost plug-and-play style un-obstructive display. The display uses an intuitive and easily understandable visualization to prompt drivers on fuel efficient velocity. The presented architecture was tested on the Idaho National Laboratory (INL) bus fleet in real-world driving conditions and was shown to be able to increase the fuel economy by 9% and 20% in two different driving scenarios.


**Abstract:** Advanced remote teleoperation of robot manipulators enable complex tasks to be performed in hostile or inaccessible environments, without the physical presence of a human. For increased effectiveness of teleoperation, maintaining accuracy and speed of task while minimizing collisions is important. Visual and auditory inputs to the user aid in accurate control. However, to further increase the speed and accuracy, tactile and kinesthetic force-feedback information can be used. One of the most common methods of force-feedback generation is the virtual force field based method. However, in complex environments where increased accuracy is required, static force field based methods are insufficient. This paper presents a dynamically varying, virtual force field based force-feedback generation method for obstacle avoidance in remotely operated robot manipulators. The presented method utilizes a fuzzy logic model to dynamically vary a virtual force field surrounding the manipulator in real-time. The fuzzy controller utilizes the distance vectors to obstacles and the velocity vectors of the manipulator components to generate the force field in each axis. The generated force field is then used to calculate the final force-feedback that is sent to the user. The presented method was implemented on a simple 3-DOF robot manipulator, and compared to a typical static force field based force-feedback generation method. Test results show that the task completion time is significantly improved without significant loss of accuracy in certain tasks when the presented force-feedback method is used.


**Abstract:** As critical and sensitive systems increasingly rely on complex software systems, identifying software vulnerabilities is becoming increasingly important. It has been suggested in previous work that some bugs are only identified as vulnerabilities long after the bug has been made public. These bugs are known as Hidden Impact Bugs (HIBs). This paper presents a hidden impact bug identification methodology by means of text mining bug databases. The presented methodology utilizes the textual description of the bug report for extracting textual information. The text mining process extracts syntactical information of
the bug reports and compresses the information for easier manipulation. The compressed information is then utilized to generate a feature vector that is presented to a classifier. The proposed methodology was tested on Linux vulnerabilities that were discovered in the time period from 2006 to 2011. Three different classifiers were tested and 28% to 88% of the hidden impact bugs were identified correctly by using the textual information from the bug descriptions alone. Further analysis of the Bayesian detection rate showed the applicability of the presented method according to the requirements of a development team.


Abstract: Fuzzy Logic Systems (FLS) are a well-documented proven method for various applications such as control classification and data mining. The major advantage of FLS is the use of human interpretable linguistic terms and rules. In order to capture the uncertainty inherent to linguistic terms, Fuzzy Membership Functions (MF) are used. Therefore, membership functions are essential for improving the understandability of fuzzy systems. Optimizing FLS for improved accuracy in terms of classification or control can reduce the understandability of fuzzy MFs. Expert knowledge can be used to derive MFs, but it has been shown that this might not be optimal, and acquiring expert knowledge is not trivial. Therefore, this paper presents a data driven method using statistical methods to generate membership functions that describe the data while maintaining the understandability. The presented method calculates key points such as membership function centers, intersections and slopes using data driven statistical methods. Furthermore, the presented method utilizes several understandability metrics to adjust the generated MFs. The presented method was tested on several benchmark datasets and a real-world dataset and was shown to be able to generate MFs that describe the dataset, while maintaining high levels of understandability.


Abstract: Fuzzy Logic Systems (FLS) are a well-documented proven method for various applications such as control classification and data mining. The major advantage of FLS is the use of human interpretable linguistic terms and rules. In order to capture the uncertainty inherent to linguistic terms, Fuzzy Membership Functions (MF) are used. Therefore, membership functions are essential for improving the understandability of fuzzy systems. Optimizing FLS for improved accuracy in terms of classification or control can reduce the understandability of fuzzy MFs. Expert knowledge can be used to derive MFs, but it has been shown that this might not be optimal, and acquiring expert knowledge is not trivial. Therefore, this paper presents a data driven method using statistical methods to generate membership functions that describe the data while maintaining the understandability. The presented method calculates key points such as membership function centers, intersections and slopes using data driven statistical methods. Furthermore, the presented method utilizes several understandability metrics to adjust the generated MFs. The presented method was tested on several benchmark datasets and a real-world dataset and was shown to be able to generate MFs that describe the dataset, while maintaining high levels of understandability.
Abstract: With escalating fuel prices, limited fossil fuel reserves and increasing carbon emissions, significant efforts are being made to decrease fuel consumption in vehicles. While, drivetrain improvements play a major role in improving fuel economy, it has been identified that fuel efficient driving behavior is a viable method for increased fuel efficiency. Thus, if the optimal fuel efficient behavior can be identified, it can be used to increase the fuel efficiency of drivers. However, once the optimal fuel efficient behavior is identified, it has to be presented to the driver, while the vehicle is being driven. Thus, this method of information representation has to be un-obstructive and easy to comprehend. This paper presents a low cost framework and a hardware setup for prompting drivers on fuel efficient behavior. The presented framework includes an information rich, intuitive un-obstructive visualization. The presented method was implemented using low cost, commercial-off-the-shelf hardware and tested on a sample of buses selected from the Idaho National Laboratory (INL) bus fleet. Different types of visual cues were and evaluated by professional drivers for obstructiveness, interpretability and intuitiveness.


Abstract: Brain Computer Interfaces (BCI) have gained significant interest over the last decade as viable means of human machine interaction. Although many methods exist to measure brain activity in theory, Electroencephalography (EEG) is the most used method due to the cost efficiency and ease of use. However, thought pattern based control using EEG signals is difficult due two main reasons; 1) EEG signals are highly noisy and contain many outliers, 2) EEG signals are high dimensional. Therefore the contribution of this paper is a novel methodology for recognizing thought patterns based on Self Organizing Maps (SOM). The presented thought recognition methodology is a three step process which utilizes SOM for unsupervised clustering of pre-processed EEG data and feed-forward Artificial Neural Networks (ANN) for classification. The presented method was tested on 5 different users for identifying two thought patterns; “move forward” and “rest”. EEG Data acquisition was carried out using the Emotiv EPOC headset which is a low cost, commercial-off-the-shelf, noninvasive EEG signal measurement device. The presented method was compared with classification of EEG data using ANN alone. The experimental results for the 5 users chosen showed an improvement of 8% over ANN based classification.


Abstract: The lithium polycarbon mono-fluoride (LiCFx) battery is sensitive to a number of dynamic environmental conditions, such as the ambient temperature. The flat discharge profile typical of the LiCFx battery current State of Charge Indicator (SOCl) technology poses a challenge to obtaining accurate estimates using conventional, voltage-based SOCl technology. To monitor this LiCFx discharge state of charge, an Artificial Neural Network (ANN) algorithm provides a nonlinear solution. The underlying
technology is charge counting with nonlinear ANN-based compensation for temperature, rate of charge, and charge history. This paper explains the design of an ANN algorithm and its training. Measurement error for this ANN based solution is estimated to be within 2-3%. Initial cost estimates are under USD $30 in bulk quantities. Drain on the battery is less than 1.5% per year, with a 10-year shelf life for the SOCI. This paper describes the performance of a sequence of prototypes based on this ANN technology. Appropriate testing results support the accuracy estimate.


**Abstract:** Building Energy Management Systems (BEMSs) are responsible for maintaining indoor environment by controlling Heating Ventilation and Air Conditioning (HVAC) and lighting systems in buildings. Buildings worldwide account for a significant portion of world energy consumption. Thus, increasing building energy efficiency through BEMSs can result in substantial financial savings. In addition, BEMSs can significantly impact the productivity of occupants by maintaining a comfortable environment. To increase efficiency and maintain comfort, modern BEMSs rely on a large array of sensors inside the building that provide detailed data about the building state. However, due to various reasons, buildings frequently lack sufficient number of sensors, resulting in a suboptimal state awareness. In such cases, a cost effective method for increasing state awareness is needed. Therefore, this paper presents a novel method for increasing state awareness through increasing spatial resolution of data by means of data downscaling. The presented method estimates the state of occupant zones using state data gathered at floor level using Artificial Neural Networks (ANN). The presented method was tested on a real-world CO2 dataset, and compared to a time based estimation of CO2 concentration. The downscaling method was shown to be capable of consistently producing accurate estimates while being more accurate than time based estimations.


**Abstract:** Wide area monitoring, protection and control for power network systems are one of the fundamental components of the smart grid concept. Synchronized measurement technology such as the Phasor Measurement Units (PMUs) will play a major role in implementing these components and they have the potential to provide reliable and secure full system observability. The problem of Optimal Placement of PMUs (OPP) consists of locating a minimal set of power buses where the PMUs must be placed in order to provide full system observability. In this paper a novel solution to the OPP problem using a Memetic Algorithm (MA) is proposed. The implemented MA combines the global optimization power of genetic algorithms with local solution tuning using the hill-climbing method. The performance of the proposed approach was demonstrated on IEEE benchmark power networks as well as on a segment of the Idaho region power network. It was shown that the proposed solution using a MA features significantly faster convergence rate towards the optimum solution.

**Abstract:** The existing Emergency Communication System (ECS) infrastructure is becoming increasingly outdated with many members of the public moving away from landline based telecommunications and broadcast television in favor of cellular telephones and internet-based streaming entertainment services. Current systems for public services such as E911 and Emergency Alert System broadcasts are no longer a reliable means for reaching the public. In addition, both wired and wireless telecommunications systems can become overwhelmed, as was the case following Hurricane Katrina in 2005 and the World Trade Center disaster in 2001, and in fact, when communications are needed most urgently, the difficulty of maintaining effective communication increases exponentially. While the use of Internet based alternatives could resolve some of these problems, existing Internet infrastructure offers no dedicated or priority bandwidth to the user for emergency communications (e.g. E911 or Emergency Alert System). The current Internet capacity can also be overloaded due to high volume network data streams. Under these conditions, emergency communications (e.g. inbound and outbound communications reporting catastrophic or emergency events) may have their packets dropped resulting in incomplete and/or delayed communications. To alleviate these problems, this paper presents a novel framework for ECS using network virtualization via Software Defined Networks (SDN). A table top demonstration of ECS using SDN was developed at the University of Idaho, Idaho Falls. This paper details the foundational technologies and overviews the steps taken at the University of Idaho to develop ECS using SDN.


**Abstract:** Significant portion of world energy production is consumed by building Heating, Ventilation and Air Conditioning (HVAC) units. Thus along with occupant comfort, energy efficiency is also an important factor in HVAC control. Modern buildings use advanced Multiple Input Multiple Output (MIMO) control schemes to realize these goals. However, since the performance of HVAC units is dependent on many criteria including uncertainties in weather, number of occupants, and thermal state, the performance of current state of the art systems are sub-optimal. Furthermore, because of the large number of sensors in buildings, and the high frequency of data collection, large amount of information is available. Therefore, important behavior of buildings that compromise energy efficiency or occupant comfort is difficult to identify. This paper presents an easy to use and understandable framework for identifying such behavior. The presented framework uses human understandable knowledge-base to extract important behavior of buildings and present it to users via a graphical user interface. The presented framework was tested on a building in the Pacific Northwest and was shown to be able to identify important behavior that relates to energy efficiency and occupant comfort.

**Abstract:** Selecting the optimal dimensions for various knowledge extraction applications is an essential component of data mining. Dimensionality selection techniques are utilized in classification applications to increase the classification accuracy and reduce the computational complexity. In text classification, where the dimensionality of the dataset is extremely high, dimensionality selection is even more important. This paper presents a novel, genetic algorithm based methodology, for dimensionality selection in text mining applications that utilizes information gain. The presented methodology uses information gain of each dimension to change the mutation probability of chromosomes dynamically. Since the information gain is calculated a priori, the computational complexity is not affected. The presented method was tested on a specific text classification problem and compared with conventional genetic algorithm based dimensionality selection. The results show an improvement of 3% in the true positives and 1.6% in the true negatives over conventional dimensionality selection methods.


**Abstract:** Brain Computer Interfaces (BCI!) are becoming increasingly studied as methods for users to interact with computers because recent technological developments have led to low priced, high precision BCI devices that are aimed at the mass market. This paper investigates the ability for using such a device in real world applications as well as limitations of such applications. The device tested in this paper is called the Emotiv EPOC headset, which is an electroencephalograph (EEG) measuring device and enables the measuring of brain activity using 14 strategically placed sensors. This paper presents: 1) a BCI framework driven completely by thought patterns, aimed at real world applications 2) a quantitative analysis of the performance of the implemented system. The Emotiv EPOC headset based BCI framework presented in this paper was tested on a problem of controlling a simple differential wheeled robot by identifying four thought patterns in the user: “neutral”, “move forward”, “turn left”, and “turn right”. The developed approach was tested on 6 individuals and the results show that while BCI control of a mobile robot is possible, precise movement required to guide a robot along a set path is difficult with the current setup. Furthermore, intense concentration is required from users to control the robot accurately.


**Abstract:** General Type-2 Fuzzy Logic Systems (GT2 FLSs) are an extension to Type-1 (T1) FLS where at least one Fuzzy Set (FS) is a GT2 FS. However, due to the high computational complexity of operations on GT2 FSs, GT2 FLSs have been rarely used in practical applications. Instead, Interval Type-2 (IT2) FLSs which employ constrained IT2 FSs, have been widely used. Despite their superior computational complexity, IT2 FLSs lack the expressive power of GT2 FSs when describing various sources of uncertainty. Further, it is unclear how to derive an IT2 FLS from a specific GT2 FLS. To alleviate these issues, this
The paper outlines a novel concept of Shadowed Type-2 Fuzzy Logic Systems (ST2 FLS). The ST2 FLS consists of previously proposed ST2 FSs, which are T2 FSs with secondary membership functions represented as Shadowed Sets (SSs). Because ST2 FSs are directly induced by GT2 FSs, the entire design of the ST2 FLS can be automatically derived from a specific GT2 FLS. Furthermore, the proposed ST2 FLS was shown to approximate GT2 FLS more accurately compared to IT2 FLS, while maintaining the computational efficiency of IT2 FLS.


**Abstract:** In the past several decades Building Energy Management Systems (BEMSs) have become vital components of most modern buildings. BEMSs utilize advanced microprocessor technology combined with extensive sensor data collection and communication to minimize energy consumption while maintaining high human comfort levels. When properly tuned and operated, BEMSs can provide significant energy savings. However, the complexity of the acquired sensory data and the overwhelming amount of presented information renders them difficult to adjust or even understand by responsible building managers. This inevitably results in suboptimal BEMS operation and performance. To address this issue, this paper reports on a research effort that utilizes Computational Intelligence techniques to fuse multiple heterogeneous sources of BEMS data and to extract relevant actionable information. This actionable information can then be easily understood and acted upon by responsible building managers. In particular, this paper describes the use of anomaly detection algorithms for improving the understandability of BEMS data and for increasing the state-awareness of building managers. The developed system utilizes modified nearest neighbor clustering algorithm and fuzzy logic rule extraction technique to automatically build a model of normal BEMS operations and detect possible anomalous behavior. In addition, linguistic summaries based on fuzzy set representation of the input values are generated for the detected anomalies which increase the understandability of the presented results.


**Abstract:** Data mining methods are becoming vital as the amount and complexity of available data is rapidly growing. Visual data mining methods aim at including a human observer in the loop and leveraging human perception for knowledge extraction. However, for large datasets, the rough knowledge gained via visualization is often times not sufficient. Thus, in such cases data summarization can provide a further insight into the problem at hand. Linguistic descriptors such as linguistic summaries and linguistic rules can be used in data summarization to further increase the understandability of datasets. This paper presents a Visual Linguistic Summarization tool (VLS-SOM) that combines the visual data mining capability of the Self-Organizing Map (SOM) with the understandability of linguistic descriptors. This paper also presents new quality measures for ranking of predictive rules. The presented data mining tool enables users to 1) interactively derive summaries and rules about interesting behaviors of the data visualized though the SOM, 2) visualize linguistic descriptors and visually assess the importance of generated summaries and rules. The data mining tool was tested on two benchmark problems. The tool was helpful in identifying important
features of the datasets. The visualization enabled the identification of the most important summaries. For classification, the visualization proved useful in identifying multiple rules that classify the dataset.


Abstract: Identifying software vulnerabilities is becoming more important as critical and sensitive systems increasingly rely on complex software systems. It has been suggested in previous work that some bugs are only identified as vulnerabilities long after the bug has been made public. These vulnerabilities are known as hidden impact vulnerabilities. This paper discusses existing bug data mining classifiers and present an analysis of vulnerability databases showing the necessity to mine common publicly available bug databases for hidden impact vulnerabilities. We present a vulnerability analysis from January 2006 to April 2011 for two well-known software packages: Linux kernel and MySQL. We show that 32% (Linux) and 62% (MySQL) of vulnerabilities discovered in this time period were hidden impact vulnerabilities. We also show that the percentage of hidden impact vulnerabilities in the last two years has increased by 53% for Linux and 10% for MySQL. We then propose a hidden impact vulnerability identification methodology based on text mining classifier for bug databases. Finally, we discuss potential challenges faced by a development team when using such a classifier.


Abstract: Data mining techniques are becoming indispensable as the amount and complexity of available data is rapidly growing. Visual data mining techniques attempt to include a human observer in the loop and leverage human perception for knowledge extraction. This is commonly allowed by performing a dimensionality reduction into a visually easy-to-perceive 2D space, which might result in significant loss of important spatial and topological information. To address this issue, this paper presents the design and implementation of a unique 3D visual data mining framework - CAVE-SOM. The CAVE-SOM system couples the Self-Organizing Map (SOM) algorithm with the immersive Cave Automated Virtual Environment (CAVE). The main advantages of the CAVE-SOM system are: i) utilizing a 3D SOM to perform dimensionality reduction of large multi-dimensional datasets, ii) immersive visualization of the trained 3D SOM, iii) ability to explore and interact with the multi-dimensional data in an intuitive and natural way. The CAVE-SOM system uses multiple visualization modes to guide the visual data mining process, for instance the data histograms, U-matrix, connections, separations, uniqueness and the input space view. The implemented CAVE-SOM framework was validated on several benchmark problems and then successfully applied to analysis of wind-power generation data. The knowledge extracted using the CAVE-SOM system can be used for further informed decision making and machine learning.
REFERENCES


[139] C. Wagner, H. Hagras, “An approach for the Generation and Adaptation of zSlices based General Type-2 Fuzzy Sets from Interval Type-2 Fuzzy Sets to Model


VITAE

Dumidu Wijayasekara received his B.Sc. in Computer Science from University of Peradeniya in Sri Lanka in 2009 and his M.Sc. in Computational Intelligence from the University of Idaho at Idaho Falls, Idaho, USA in 2014. He joined the Doctor of Philosophy program at Virginia Commonwealth University in Richmond, Virginia in 2014. His research experience includes research assistant positions at the University of Idaho and the Virginia Commonwealth University. His fields of interest include fuzzy systems, machine learning, pattern recognition, data mining, human machine interaction, and advanced visualization systems.