Fuzzy Transfer Learning

Jethro Shell, B.A.(Hons.), M.Sc.

Submitted in partial fulfilment
of the requirements for the degree of Doctor of Philosophy.

Faculty of Technology

De Montfort University

June 2013

Acknowledgements

I would firstly like to thank Dr. Simon Coupland for his unwavering support through out this process alongside the rest of my supervisory team, Dr. Chi-Biu Wong and Dr. Eric Goodyer who assisted in supporting my endeavours.

I would also like to thank those members, past and present, within the Centre for Computational Intelligence at De Montfort University who offered assistance with this project, both technical and inspirational. A special mention goes to Prof. Bob John for financial and academic assistance, and Dr. Stephen Matthews, Dr. Hussam Hamrawi, Dr. Benjamin Passow and Dr. Stephen Vickers for their overall support.

Special thanks goes to my parents, Gayle and Michael Shell alongside my brother Jacob and his family. All have have been life long inspirations and have offered unbounded support. The greatest thanks, however, goes to my wife Claire, who has been a rock of support through out from the onset. Without her unshakeable strength, love, and endless encouragement I would not have been able to complete this research. To her, I will be eternally grateful.

Abstract

The use of machine learning to predict output from data, using a model, is a well studied area. There are, however, a number of real-world applications that require a model to be produced but have little or no data available of the specific environment. These situations are prominent in Intelligent Environments (IEs). The sparsity of the data can be a result of the physical nature of the implementation, such as sensors placed into disaster recovery scenarios, or where the focus of the data acquisition is on very defined user groups, in the case of disabled individuals.

Standard machine learning approaches focus on a need for training data to come from the same domain. The restrictions of the physical nature of these environments can severely reduce data acquisition making it extremely costly, or in certain situations, impossible. This impedes the ability of these approaches to model the environments. It is this problem, in the area of IEs, that this thesis is focussed.

To address complex and uncertain environments, humans have learnt to use previously acquired information to reason and understand their surroundings. Knowledge from different but related domains can be used to aid the ability to learn. For example, the ability to ride a road bicycle can help when acquiring the more sophisticated skills of mountain biking. This humanistic approach to learning can be used to tackle real-world problems where *a-priori* labelled training data is either difficult or not possible to gain. The transferral of knowledge from a related, but differing context can allow for the reuse and repurpose of known information.

In this thesis, a novel composition of methods are brought together that are broadly based on a humanist approach to learning. Two concepts, Transfer Learning (TL) and Fuzzy Logic (FL) are combined in a framework, Fuzzy Transfer Learning (FuzzyTL), to address the problem of learning tasks that have no prior direct contextual knowledge. Through the use of a FL based learning method, uncertainty that is evident in dynamic environments is represented. By combining labelled data from a contextually related source task, and little or no unlabelled data from a target task, the framework is shown to be able to accomplish predictive tasks using models learned from contextually different data.

The framework incorporates an additional novel five stage online adaptation process. By adapting the underlying fuzzy structure through the use of previous labelled knowledge and new unlabelled information, an increase in predictive performance is shown.

The framework outlined is applied to two differing real-world IEs to demonstrate its ability to predict in uncertain and dynamic environments. Through a series of experiments, it is shown that the framework is capable of predicting output using differing contextual data.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The world that surrounds us is complex, consisting of many unknowns. To interpret this world, humans have learnt to use the information that they acquire to reason and understand their surroundings. However, there exists in measuring the world a level of uncertainty, and any inference then drawn. To make decisions, analyse information, classify situations or predict events, humans can use information that encompasses them, or previous knowledge to mitigate the uncertainty. Many Computational Intelligence (CI) methodologies have approached complex and dynamic real-world problems by drawing on this humanistic learning.

The availability of information produces differing understanding of problems. Knowledge is composed of sourced information and the understanding that is subsequently ascertained. A lack of information reduces the ability to understand a problem. Differences in information and understanding about a problem domain can be defined as being contained within a *knowledge gap*. This thesis presents a novel composition of methods, broadly based on a humanist approach to learning, to bridge the knowledge gap. Two concepts, Transfer Learning (TL), a methodology that allows information gained in different contextual situations to assist new learning tasks[1], and Fuzzy Logic (FL), an approach to capture imprecision and uncertainty, are brought together in a novel framework to address the problem of learning tasks that have no prior direct contextual knowledge.

Real-world applications often consist of many unknowns. To predict or classify based on the information gathered from these applications can be extremely difficult. Standard machine learning scenarios require that there is a form of training data. Predominantly there is a requirement for such training data to come from the same domain. Some applications make the procurement of *a priori* labelled training data extremely difficult, or in some cases, not possible at all. For example, to measure certain physical areas such as remote forest locations, impromptu set ups such as disaster zones, or small user groups that have very defined requirements such as disabled users.

[1] In this research the term *task* is referred to as any action the learning method is required to accomplish

The procurement of training data produces an interesting problem. If there is a requirement to classify or predict the output from such environments, *how can a model be produced?* The examples given previously present situations where labelled data from the same distribution may be extremely difficult to acquire. Additionally, large quantities of unlabelled data may also not be available. Within this situation, standard supervised, semi-supervised and unsupervised learning strategies are not applicable. This thesis focusses on these situations.

The contexts discussed can, however, be related to other implementations which may contain previously discovered knowledge. The transferral of knowledge from one context to another is in keeping with the concept of a more humanist style of learning, to reuse and repurpose information. When a human is faced with a new, unknown task they often rely on previous experience to solve the problem or answer questions that are raised. If the task is closely related to one encountered before, the ability to solve the task or answer the questions may become easier.

Within the study of human learning, ordinary learning is viewed as being ordinary when it is within the same context (a student may solve similar problems that are at the end of a chapter that have appeared previously), whereas TL occurs outside of a single context (problems are solved when they occur mixed with others at the end of the course) (Perkins & Salomon 1992). Studies have shown that humans often draw upon more than just training data for generalisation (Thrun 1996). In recent years there has been significant quantities of research in the area of TL and its application to real-world problems in the area of CI (Xu & Yang 2011, Gorski & Laird 2006, Hu & Yang 2011, Hu & Yang 2011, Barrett et al. 2010). TL can be broadly defined as a learning technique that uses knowledge from a source domain to increase the performance of learning within the target task domain. The methodology allows the domains, tasks and distributions used within the training and testing to be different. The research within this thesis presents a novel use of a TL method to model scenarios where little or no information is initially known.

There is a strong relationship between context and uncertainty. As individuals endeavour to learn a new task they often afford uncertainty to it. There is a clear codependency on the level of certainty in any learning activity and the amount of information that is available. Problems with little information can have a high degree of uncertainty (Mendel 1995). The lack of knowledge that is manifested as real world problems are addressed is an embodiment of uncertainty. Imprecision, approximation, vagueness and ambiguity of information are driven by the variability encountered when trying to measure the world. Dynamic applications such as Intelligent Environments (IEs) can exhibit this uncertainty in the sensors that are used and the decision structures that are applied. In this thesis, the incorporation of a fuzzy logic system is proposed to assist in the modelling of environments in presence of uncertainty and vagueness. The use of fuzzy logic allows for the incorporation of approximation and a greater expressiveness of the uncertainty within the data (Zadeh 1988).

1.1 Thesis Summary

To summarise, this thesis presents a novel framework, Fuzzy Transfer Learning (FuzzyTL), that uses the methods within FL and TL to bridge the *knowledge gap* between the learning process of one context to another. Whilst the abilities of the framework have been shown to be applied to predictive tasks (as illustrated in Chapter 4), there is a belief that the generic nature of the framework allows it to be applied to problem spaces beyond these confines. The novel methods and the application of those techniques have been previously presented in a number of conference papers (Shell et al. 2012, Shell & Coupland 2012). For clarity, these can be found in the Appendix.

A further discussion of the motivation for the proposed framework is given in the following section.

1.2 Motivation

The prime motivation for the FuzzyTL methodology can be summarised by presenting a simplified example. This example is based on the application of environmental control in IEs. The IEs are represented by two separate residential homes constructed using sensors to monitor various environmental controls. ***Home A*** is a residential flat with three rooms containing nine sensors (occupancy, temperature and heating activation in each room). The home is occupied by a single resident. Data is recorded during the month of March. A second flat, ***Home B*** with five rooms has a reduced number of sensors containing only occupancy and temperature sensors in each room. This home is occupied by a couple. This implementation was configured at the beginning of September of the same year as ***Home A***. The structure of these residences can be seen in Fig 1.1.

Within the Intelligent Environment (IE) proposed, the heating system is automated. There has been an increasing quantity of research that has focussed on the automation of environmental control (Mozer 1998, Scott et al. 2010, Wagner & Hagras 2010). Such systems have included the control of heating and air conditioning activation to maintain the desired environment. In order to understand when to turn on a heating system, open a window or activate the air conditioning a model needs to be created of the environment. Focussing on the example provided, the data collected from the occupancy, temperature and heating activation over a period of time can be used to generate such a model. Conditions both outside and inside of the home can lead, however, to variation and uncertainty in the data that is collected. A variable such as the outdoor temperature can influence not only the temperature within the home directly but the reaction of the occupants. Residents can react to decreases in the outside temperature by activating the heating. The number of occupants may also have an influence. Variation in types of activity will have an impact on the values the system may record. These factors demonstrate that to produce a model in such an environment is complex, as a high number of imprecise, uncertain variables can be involved.

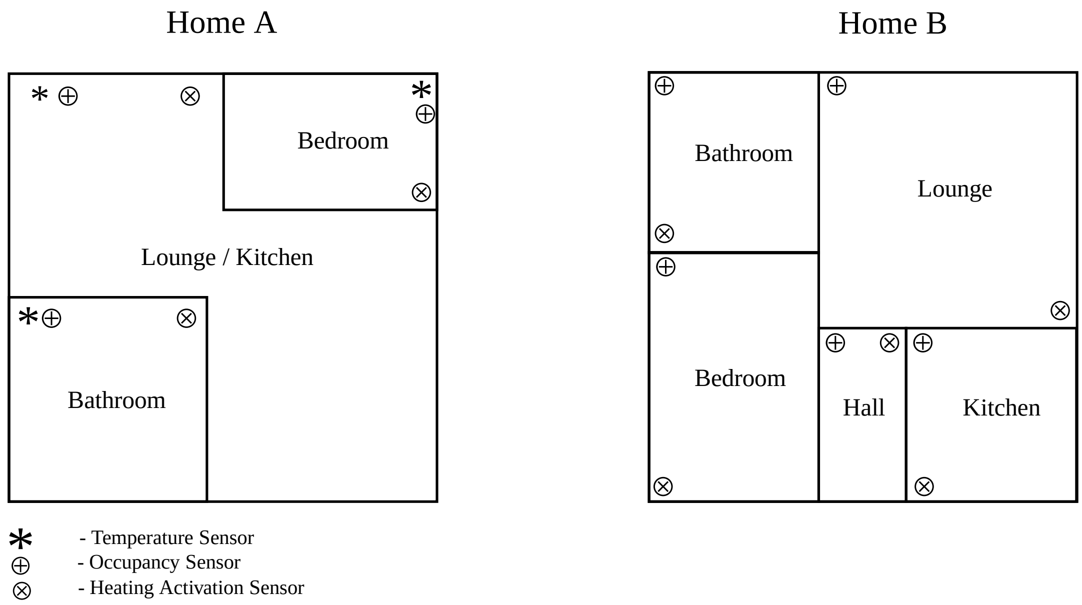

Figure. 1.1. Floor Plan of Home A and Home B

Using a standard machine learning process, the data that is produced from ***Home A*** would allow for the production of a model to predict when it was necessary to activate the heating system. This can be induced using a supervised learning method from the the data that has been supplied from the sensor structure. The production of a model for the heating activation for ***Home B*** is more difficult. There is no prior data regarding the heating activation output on which to base it. There is possible scope for the use of models created from differing contexts.

In classical supervised, semi-supervised and unsupervised learning approaches, the application of the same model across different contextual tasks such as across ***Home A*** and ***Home B*** and between March and September, would require the production of a new model specific to the domain and feature space. To varying extents, each learning approach requires a set of training data in order to construct a model. Unsupervised learning methods use unlabelled data to produce the model. The lack of annotated data implies that the model is derived from the input data itself. Techniques such as clustering, novelty detection and dimensionality reduction are used within this field (Zhu & Goldberg 2009). Semi-supervised is broadly a learning method that uses a large amount of unlabelled data combined with a small amount of labelled data to build a model. This approach is often used in cases when obtaining unlabelled data is cheap or easy, while labelled data is expensive or difficult (Chapelle et al. 2006). Supervised learning alternatively uses labelled data to produce a model. Based on the labelled data provided within the training data set, a function is created that can predict the output values mapped to input data within the feature space. These techniques typically require the distributions of the data to be within the same domain in order to produce an effective output.

Taking the outlined example, despite the perceived similarities between ***Home A*** and ***Home B***, the data distributions may vary. If within ***Home A*** the temperature significantly rose in the last week of March, this would change the range of this input variable across the domain. Using the labelled data as a training source, each learning methods would produce a model based upon these results. If the temperature during September within ***Home B*** varied far less and over a lower range, the model may become imprecise.

FuzzyTL addresses the problem of learning a model where there is no labelled data, and initially extremely sparse unlabelled data, such as IE example previously given. This thesis focuses on modelling and predicting output from IEs as they represent dynamic, real-world applications, often producing uncertain and vague data. There is a requirement in this area to model ad-hoc (George et al. 2010), remote (Werner-Allen et al. 2005) and highly varied domains using limited knowledge.

The rest of this thesis outlines the FuzzyTL framework and its ability to use differing contextual information to predict output. To test the novel methodology presented, hypotheses are initially constructed.

1.3 Hypotheses

Two hypotheses will be tested in the course of this thesis.

Hypothesis 1: Where minimal unlabelled data is available within a target task, data in the form of a TL process from contextually related but differing source tasks, can be used to learn predictive tasks.

A series of experiments are presented in Chapter 4 demonstrating the ability of the FuzzyTL framework to predict sensor values. The experiments are based on real-world, dynamic IE data that contains noisy and uncertain information. An evaluation of the performance of the FuzzyTL framework was calculated through the comparison of the predicted value and known sensor readings alongside a benchmark dataset. Source data was provided from different contexts along with an increasing quantity of unlabelled target data. The FuzzyTL is shown to perform well, absorbing contextual changes.

Hypothesis 2: Adaptation of the transferred source domain through the use of unlabelled new data can increase the performance of FuzzyTL in predicting target tasks.

The information that is contained within the unlabelled target data is used in the FuzzyTL framework to enhance the TL through an online adaptation of the Fuzzy Inference System (FIS). Chapter 3 describes this novel methodology in detail. Experiments carried out in

Chapter 4 confirm that overall the use of online adaptation increases the performance of the FuzzyTL framework in predicting target tasks.

1.4 Major Contributions of the Thesis

The contributions of this thesis can be summarised as following:

1. A novel framework for the learning of target tasks from limited unlabelled target data and related, differing source labelled data using a FIS.

2. A novel adaptive online learning methodology for the use with limited unlabelled data to enhance the transfer of a FIS between contextually differing learning tasks.

3. A novel addition is provided to the Wang-Mendel (WM) method for the learning of fuzzy rules from numerical data using a fuzzy frequency approach.

4. The first application of a FuzzyTL framework on IE datasets to perform predictive learning tasks.

1.5 Structure of the Thesis

The structure of this thesis will be as follows:

Chapter 2 introduces all of the required literature and background information that is needed to understand the following chapters. Firstly, an introduction is given to the concept of context, its application to this work and its relationship to uncertainty. A definition is given within this section that is used throughout the thesis. The following section describes the use of Fuzzy Logic (FL), initially outlining the interaction of uncertainty and the FL methodology. Included within this section is a discussion of the methods used within the Fuzzy Transfer Learning (FuzzyTL) framework for fuzzy rule extraction. The employed method is described in detail along with other methodologies in order to compare the attributes of each system. The section culminates with a review of the application of FL in the key area of Intelligent Environments (IEs). This assists in setting the scene for the implementation of the experimental structure and application of the defined methodology.

Chapter 3 forms the major contribution of this thesis. Within this chapter the novel FuzzyTL framework is described in detail. The initial learning stage is defined incorporating the addition of the novel fuzzy frequency rule pruning. The transfer of the fuzzy model is discussed giving an insight into the frameworks overall structure. The final section describes the five stage adaptation process that incorporates both the adjustment of the fuzzy set structure and the adaptation of the rule base.

Chapter 4 shows the application of the FuzzyTL framework on two IEs. The chapter firstly gives details of the formation of the data used in the application of the framework. Two data sets are used in the experimentation demonstrated in this chapter, each of the data sets representing differing contextual situations. Each experimental structure is described and discussed in detail. The chapter is concluded by a summary of the findings and discussion of the results.

Chapter 5 presents a discussion and concluding summary of the research presented in the thesis. The major findings of the thesis are discussed with an overall summary of the contributions given. This chapter also includes a discussion of recommendations and possible areas of future work.

Chapter 2

Literature Review

2.1 Introduction

The work in this thesis presents a novel methodology, Fuzzy Transfer Learning (FuzzyTL). Within this chapter a discussion will be given of the fundamental elements that underpin the methodology and introduce the application areas used. The framework draws on two learning methods, Transfer Learning (TL) and Fuzzy Logic (FL). In Section 2.2, a discussion is given of context with its relationship to both FL and TL. The understanding of context within the area of computing is vital to the way in which the FuzzyTL framework is implemented. A definition of context is also given within this section. Section 2.3 gives an overview of FL offering a background to the terminology that will be used in this thesis and the fundamental techniques that are used to manage uncertainty within the framework. An overview of the Ad-Hoc Data Driven Learning (ADDL) methodology applied to the framework discussed in this thesis will be given with a discussion of comparable methods. To assess the function of the TL components of the proposed methodology, Section 2.4 gives an overview of the process. A discussion is supplied to demonstrate the applicability of combining Computational Intelligence (CI) methods with TL to solve differing real-world problems. Finally, Section 2.5 puts forward previous work in the area of FL and Intelligent Environment (IE). As the main experimental work of this research is focussed on IEs, this section establishes the use of FL in managing uncertainty within such environments.

2.2 Context

The concept of context plays an important role in both FL and TL, however there is no single consensus of how context should be defined. In the following section a number of views will be discussed of application and implementation before a definition will be proposed. Finally a brief discussion will be presented of uncertainty and its relation to context.

2.2.1 Applications and Implementations of Context

Research in the area of context and Context Aware (CA) computing has grown in parallel with the emergence of technologies such as ubiquitous computing and the semantic web. Traditional Human Computer Interface (HCI) has moved away from the constrained environment of a single computer at a desk or interaction based on a screen and a keyboard (Dourish 2004). Mobile devices, sensor networks and what has become defined as the *internet of things* allows for a multitude of differing interactions. The everyday computing context has changed. Computing contexts are no longer static or well defined, but are often vague and uncertain.

The structure of the FuzzyTL framework has foundations in the notion of context. As illustrated later in Section 2.4, TL has the ability to use information from one domain to close the information gap in a learning process from differing but similar domains[1] The domains can be defined as contexts. To analyse the contexts, a valid definition of a context must be put forward.

There have been many template definitions of context, of which three are discussed here, that focus on CA computing. Schilit et al.(1994) discuss three important aspects of context in relation to mobile computing: where you are, who you are with, and what resources are nearby. However, they expand this to include: lighting, noise level, network connectivity, communication costs, communication bandwidth, and even the social situation. Taken in its most abstract form, Schilit's interpretation still allows for a tangible, measurable definition. It focusses on the individual within a domain and their interaction with the world around them. Context-aware computing focuses strongly on the activity of a user and the environment in which they are surrounded. Jang (Jang 2005) proposed a unified model for a context-aware system. Jang's system puts forward the idea of the independence of a sensor from the application in terms of 5W1H (Who, What, Where, When, How, and Why). According to Jang, most context-aware systems provide data as part of the 5W1H system such as user identity, location and time. Based upon this, a unified 5W1H model is believed to work for most systems without loss of generality (Jang 2005).

Dey (Dey 2001, p2) puts forward a definition of context:

> "Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

The definitions of Jang and Dey relate to Dourish's representational view of context (Dourish 2004). Dourish uses the representational nature of software systems to represent and encode context. The definitions of Schilit et al. (Schilit et al. 1994), Dey (Dey 2001) and Jang (Jang 2005) describe context through its relationship to information which is formed or expressed, to varying

[1] The use of domain is within the context of TL. This is examined further in Section 2.4.1.

degrees of abstraction. Dourish asserts four assumptions regarding context that are based upon these types of definition. They are:

1. *Context is a form of information.* It is something that can be known and therefore represented.
2. *Context is delineable.* For an application, or set of applications, the context of the activities which the application supports can be defined.
3. *Context is stable.* Variation may occur within elements of the application from application to application, however they do not vary from instance to instance.
4. *Context and activity are separable.* Context describes the features of an environment that an activity occurs within.

Relating to implementations in context-aware Intelligent Environments (IEs), Meyer and Rakotonirainy (Meyer & Rakotonirainy 2003) discussed that context can refer to the circumstances or situations in which a computing task takes place. The context of an entity is any measurable and relevant information that can affect the behaviour of the same entity. This is a broad and abstract definition, however Meyer and Rakotonirainy's definition has similarities to Dourish's work. A more focussed definition came from Elnahrawy and Nath (Elnahrawy & Nath 2004). They proposed to the use contextual information to identify missing sensor values, and anomalies or malicious sensor readings. Their approach is based on exploiting the spatio-temporal relationships that exist among sensors in WSN's.

2.2.2 Defining Context

For the purposes of this thesis, a high level abstract definition of context will be used. Taking influence from the work of Dey (Dey 2001), Dourish (Dourish 2004) and Bettini et al. (Bettini et al. 2010), this thesis defines context as:

1. **Information:** Each context consists of definable variables that are relevant and measurable.
2. **Behaviour:** The context embodies an entity, application, service or group thereof that is affected by the behaviour of the associated information.
3. **Variation:** Differences within the structure of the variables can occur between context to context, but not from instance to instance within a context itself. This would be defined as a new context.

2.2.3 Uncertainty in Context

By defining a contextual domain, modelling is allowed to occur. Most studies of contexts and context-aware computing are focussed on the measuring of the real, physical world. Such measurements are prone to uncertainty and imprecision. One of the key requirements of studying contexts is capturing and understanding of imprecise and possibly conflicting data about the physical world (Bettini et al. 2010). A number of studies have endeavoured to address the problem of uncertainty in context information. Dey (Dey et al. 2000) suggests that contextual uncertainty can be resolved by a mediation process that involves interaction with the user. Dámian-Reyes et al. (Damin-Reyes et al. 2011) discussed the use of an Uncertainty Management Heuristic Mechanism (UMHM). This applies a three-phase approach to manage uncertainty. Possible sources of uncertainty are identified and represented before determining how to proceed. Ranganathan et al. (Ranganathan et al. 2004) developed an uncertainty model based on a predicate representation of contexts combined with a confidence value. The predicate representation follows a convention of naming the type of context being described, for example, location or time. Some contexts are considered to be more certain than others. A structure such as an office may be certain because locality is well defined, whereas time is less so. A confidence value is attached to each predicate. The value measures the probability or membership value that the event is true.

2.2.4 Discussion

This section discussed a general view of context alongside the application of context within ubiquitous computing. The implementations that are used to demonstrate the frameworks abilities are reliant on understanding context. To these ends, the definition of context proposed within this section will be used throughout the rest of this thesis. Additionally, it was demonstrated that context is directly linked to the measurement of the real-world. The measuring of any application must take into account the context in which it exits. As both the real-world and subsequently the context it embodies are evaluated, the uncertainty and vagueness that are contained become more apparent. Within the following section, the use of FL to represent this uncertainty is shown to be a valid proposition.

2.3 Fuzzy Logic

There is a need to capture and effectively represent the uncertainty and vagueness that exist in real-world environments. Standard probability and logic lack the capabilities to achieve this. Zadeh (1965) introduced the concept of fuzzy sets which he later expanded on by introducing further aspects of Fuzzy Logic (FL) including fuzzy rules in (Zadeh 1973). The two primary elements within FL, the *linguistic variable* and the *fuzzy if-then rule* are able to mimic the humanistic ability to capture imprecision and uncertainty within linguistic values. FL has found favour in a broad variety of applications and in numerous, divergent Computational Intelligence (CI) incarnations. FL forms a major component of the Fuzzy Transfer Learning (FuzzyTL) framework. Through the use of FL, the imprecision that exists within real-world environments such as Intelligent Environments (IEs) can be expressed more effectively, capturing a greater degree of the information contained within the context (a greater explanation of IEs and the application of FL within IEs will be given in Section 2.5 and 2.5.2 respectively).

Initially within this section, a discussion is given of use of FL to express uncertainty. The main body provides an overview of the FL components with the inclusion of an explanation of the Fuzzy Inference System (FIS). Within the FuzzyTL framework, the FIS forms the main embodiment of the decision making system. The final section discusses methods for learning fuzzy rules and fuzzy sets including the Wang-Mendel (WM) process that is adapted and incorporated into the FuzzyTL framework.

2.3.1 Uncertainty

Much of science requires the pursuit of precision and exactness. However, humans live in a world that is formed by imprecision, vagueness and uncertainty. Real world applications are particularly at the mercy of this world. As people endeavour to measure the world, imprecision emerges. The cost associated with the pursuit of increasing precision rises in equal measure. As an example, parking a car is a simple task as it only requires that the final placement of the vehicle is imprecise. Generally, parking spaces allow for a large margin of error. Decreasing the error margin from many centimetres to only a few millimetres, and so increasing the precision, would drastically increase the cost associated in terms of execution (Zadeh 1994). Similarly, uncertainty is codependent on the quantity of information that is available. As more information about a problem is acquired, individuals become more certain about its formulation and solution. Problems with less information have a higher degree of uncertainty (Mendel 1995). The uncertainty within a problem can exist in many ways. Mendel states that:

> "Uncertainty can be manifested in many forms: it can be fuzzy (not sharp, unclear, imprecise, approximate), it can be vague (not specific, amorphous), it can be ambiguous (too many choices, contradictory), it can be of the form of ignorance (dissonant, not knowing something), or it can be a form due to natural variability (conflicting, random, chaotic, unpredictable)" (Mendel 1995).

By its very nature, uncertainty increases with a lack of knowledge. Uncertainty can be considered as existing in the *knowledge gap*. A knowledge gap can be broadly defined as the level of understanding that is exhibited based on the information that is known, compared to an optimum level of understanding. Using the previous car parking example, the optimum level of understanding may be considered to be how a driving instructor will park a vehicle having full vision of the parking bay. Comparably, a student may have a reduced level of understanding. In this scenario, information may be limited, conflicting and vague because of limited driving experience and vehicles partially obscuring the view of the parking bay. This alters the learning and understanding of the task, manifesting itself as uncertainty. The difference in the understanding of the task that the instructor and the student exhibit is expressed by the knowledge gap.

To tackle the uncertainty and vagueness that is exhibited within the real-world, the concept of soft computing has been developed. A group of methodologies that has gained increasing recognition, soft computing is focussed on using the tolerance for imprecision and partial truth to produce a system that is robust and tractable (Zadeh 1994). Soft computing is generally regarded to encapsulate three main components: neurocomputing, probabilistic reasoning which embodies methodologies such as Genetic Algorithms (GAs) and belief networks, and FL. Within this literature review, the discussion will be primarily centred on the application of FL. The components of FL were formed to capture the imprecision that is embodied within real-world applications. The framework put forward by Zadeh, principally in *Fuzzy Sets* (Zadeh 1965), and expanded upon later (Zadeh 1973), deals with the sources of imprecision.

2.3.2 Fuzzy Logic Sets

To introduce FL, classical set theory is discussed. In this thesis, the list method will be used.

$$A = \{a, b, c\} \tag{2.1}$$

Equation 2.1 shows a set A with members a, b and c. Within classical set theory, a set can be described as introducing the concept of dichotomisation to a list of objects. This can be defined with the function:

$$A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases} \tag{2.2}$$

Figure 2.1 shows the boolean set A against its membership μ.

Classical set theory requires that the boundaries of the sets are defined precisely. The membership of the sets is therefore determined with certainty (Klir et al. 1997). An item is either within the set or it is not. Most sets cannot be so well defined. This is especially true of real-world applications. The world in which we live today is imprecise, uncertain and hard to be categorical about (Zadeh 1994). This imprecision and uncertainty manifests itself in many forms. Any measurement that is taken holds uncertainty within it due to the inevitable erroneous nature

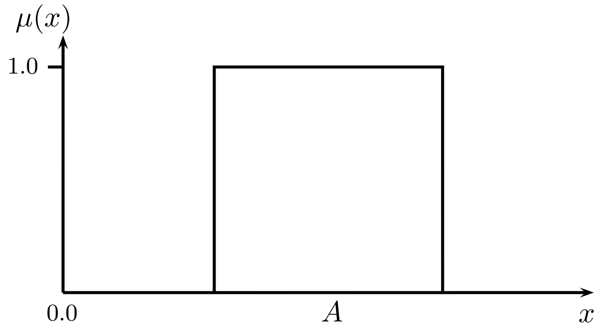

Figure. 2.1. Example of a Function as Depicted as a Classical Set.

and resolution of instrumentation (Klir & Wierman 1999). In order to capture this uncertainty and represent it effectively and efficiently, standard probability and logic structures lack the required capabilities. The move away from the crisp nature of probability emerged initially through the work of Max Black (Black 1937) and subsequently with the introduction of FL by Lotfi Zadeh (Zadeh 1965). In 1965, Zadeh introduced the concept of a fuzzy set, a set that has no crisp boundaries. A member of a fuzzy set may be inside the set to a *greater* or *lesser* degree (Klir et al. 1997). For example, a set of tall people does not fit into a classical set structure. By examining individuals, it is extremely difficult to define whether someone is tall or not. This is due to the continuous nature of the concept of height and the interpretation of linguistic terms by humans. A person can be defined as being tall at 1.80m and above, then a person at 1.79m would not be considered tall (Klir et al. 1997).

A fuzzy set can be defined as a membership function μ_x that associates with each point in x in the universe of discourse X which is a real number interval $[0, 1]$. The value of $\mu_A(X)$ represents the grade of the membership of the input value (Mendel 2000). The membership functions can be defined as the form:

$$\mu : X \to [0, 1] \tag{2.3}$$

There are a number of differing types of membership functions that are used. This thesis uses both triangular and Gaussian functions. A generic, symmetric, triangular membership function is defined as:

$$A(x) = \begin{cases} b\big(1 - \frac{|x-a|}{s}\big) & \text{when } a - s \leq x \leq a + s \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

where a is the centre of the set, b is the height and s is the width (Klir et al. 1997). A graphical representation of a triangular fuzzy set can seen be in Figure 2.2.

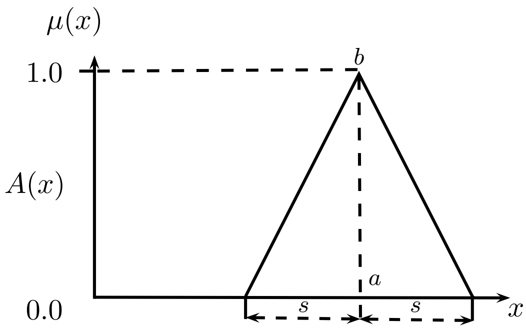

Figure. 2.2. Example of a Triangular Membership Function.

A Gaussian function takes the form of:

$$A(x) = ce^{\frac{-(x-a)^2}{b}} \tag{2.5}$$

where a is the centre of the function, c is the height and b is used to form the width.

Using the graphical representation of a triangular fuzzy set, the example of a persons height can be expressed as two fuzzy sets. These are given linguistic labels *Short* and *Tall* (see Figure 2.3). Unlike in classical set theory where values are contained solely in a single boundary, in FL an element may have a membership of more than one.

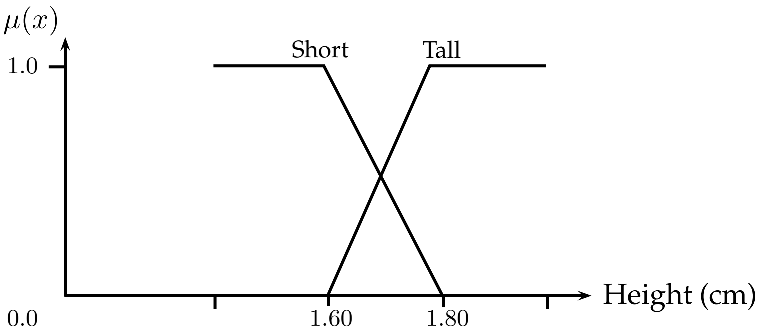

Figure. 2.3. Height of an Individual Expressed as Fuzzy Sets.

In this thesis, there is a predominant use of triangular fuzzy sets to capture and represent the information that is gathered in the data. The use of fuzzy sets and the adaptation there of, is fundamental to the FuzzyTL framework. A further discussion of fuzzy set adaptation will be given in Sections 2.3.5 and 2.3.9.

2.3.3 Fuzzy Rules

The second major aspect of FL is the fuzzy rule which are sometimes referred to as *fuzzy if-then rules*. A rule is a proposition with previously defined terms, for example in the context of a proposition applied to an Intelligent Environment (IE) the term heating is *on* could be used. In the same way, a rule is constructed as:

IF the radiator is hot **AND** the house is occupied **THEN** the heating is high.

In this form of rule structure, the **IF** statement takes the form of antecedent values and the **THEN** statement takes the form of consequent values. These can be defined as m antecedent variables $x_1, \ldots, x_m$ that are mapped to n consequent variables $y_1, \ldots, y_n$.

Fuzzy propositions can take two forms: atomic fuzzy propositions, and compound fuzzy propositions (Wang 1999). An atomic proposition is expressed as a single statement, for example x is A where x is a linguistic variable such as *heating* and A is a term such as *on*. Compound propositions are compositions of fuzzy propositions using the connections *and*, *or* and *not*. For each of the connections a specific binary operation can be carried out.

Using two linguistic variables A_1 and A_2 in X, the combinations $\bar{A}_1, A_1 \wedge A_2, A_1 \vee A_2$ are described by Mamdani (1997) as:

- A_1 *AND* A_2 is formed from $min(\mu_{A_1}, \mu_{A_2})$ as the membership value of each element of the set.
- A_1 *OR* A_2 is formed from $max(\mu_{A_1}, \mu_{A_2})$ as the membership value of each element of the set.
- *NOT* $A_1(\bar{A}_1)$ is formed from $(1 - \mu_{A_1})$ as the membership value of each element of the set.

These can be defined as fuzzy intersection (*AND*), fuzzy union (*OR*) and fuzzy complement (*NOT*). The intersection of two fuzzy sets is defined as a binary mapping which aggregates two membership functions. This can be referred to as a t-norm operator. Similarly, a fuzzy union can be represented as the addition of two membership functions. This can be represented as a binary operator, t-conorm.

There are a number of inferential processes that can be used, however within this thesis the *minimum implication* first proposed by Mamdani (Mamdani 1974) and the *product implication* proposed by Martin Larsen (Martin Larsen 1980), are used. Both implications are widely used within applications, and can be easily be implemented due to their ease of computation (Mendel 2000). Using the rule **IF** x_1 **THEN** y_1 they can be described as:

$$\mu_{(x_1,y_1)} = min[\mu(x_1), \mu(y_1)] \tag{2.6}$$

$$\mu_{(x_1,y_1)} = \mu(x_1) \cdot \mu(y_1) \tag{2.7}$$

The formation of fuzzy rules is a well studied subject. Many processes have been used to produce a fuzzy rulebase. This thesis is focussed on the use of automated means to produce fuzzy rules. A discussion regarding methods to construct a fuzzy rulebase can be found within Section 2.3.5.

Fuzzy sets and rules form two of the major components of the FIS. The FIS is an integral part of the FuzzyTL framework.

2.3.4 Fuzzy Inference System

A FIS, also referred to as a *Fuzzy Logic System*, a *fuzzy rule-based system*, a *fuzzy model* or a *fuzzy controller*, is a widely used approach for control systems that has been applied to a number of applications (Martin Larsen 1980, Lee 1990). A strength of the FIS is the ability to handle linguistic concepts and perform non-linear mapping between inputs and outputs (Guillaume 2001). FIS takes crisp inputs and maps them to crisp outputs. An FIS principally contains four components: fuzzy rules, a fuzzifier, an inference engine, and a defuzzifier (Mendel 2000). Figure 2.4 shows the components of the system. As fuzzy rules have already been discussed, each of the remaining items will be reviewed.

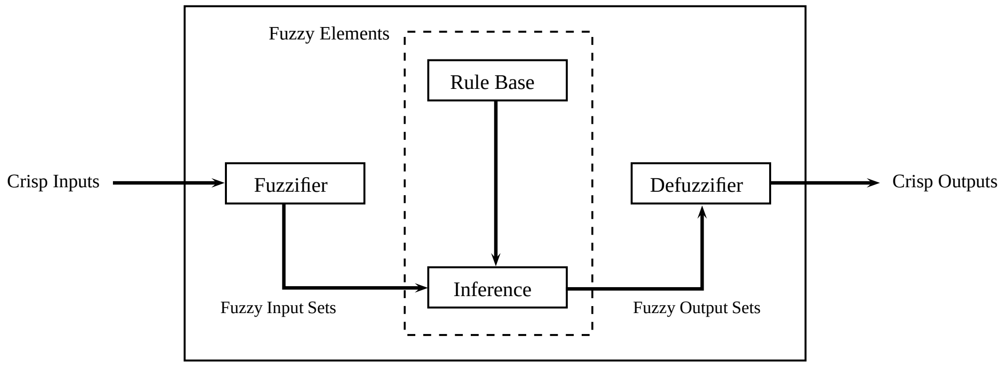

Figure. 2.4. The Structure of a Fuzzy Inference System Adapted From (Mendel 1995, Jang 1997, Lee 1990).

The fuzzifier serves as a way of mapping a crisp input to a fuzzy set. It transforms a numeric value to a fuzzy set (Roychowdhury & Pedrycz 2001). The fuzzifier also performs the function of converting the input data into suitable linguistic values which are seen as the labels of the fuzzy sets (Lee 1990).

The fuzzy inference engine uses FL operators to combine the fuzzy rules in the rulebase, mapping one fuzzy set to another. If the fuzzy rulebase only holds a single rule, then the mapping from the input to the output is direct. Rulebases almost exclusively contain more than one rule. The FL principles used to combine the fuzzy rules are the same methods that are employed within the rule construction phase. The fuzzy inference engine, however, may manage multiple antecedent values.

The final part of the FIS is to return a crisp output. Conceptually, the task of the defuzzification method is to specify a point that best represents the set that has been constructed by the fuzzification process. The defuzzifier can be defined as a mapping from the output sets to a crisp number. There are a number of methods that can be used to achieve this. For a survey of defuzzification strategies see (Roychowdhury & Pedrycz 2001). One of the most regularly used within applications is the Centre Of Gravity (COG) approach. This defuzzification method is based upon a weighted average process. The COG method can be defined as:

$$COG = \frac{\sum_{i=1}^{n} x_i \mu(x_i)}{\sum_{i=1}^{n} \mu(x_i)} \tag{2.8}$$

The design of a FIS system can fall into two main categories: the use of expert knowledge, those referred to as Fuzzy Expert Systems (FES), and those produced from data. Expert systems produce FIS with high semantic levels and good generalisation, however as the complexity increases accuracy can decrease. A goal of this thesis is to automate the production of a model of a real-world application, as a result expert systems will not be covered.

Various automated learning methods can be used to generate the main elements of a FIS, namely the fuzzy sets and the rulebase. The construction of these elements can be split into broad areas: rule induction methods, clustering, neural networks, evolutionary methodologies and Evolving Fuzzy Systems. Each learning method approaches one or a number of these areas of the FIS. In the following section, an overview of a number of learning methods will be given with a focus on Rule Induction Methods.

2.3.5 Rule Induction Methods

The production of rules by inductive methods allows for the extraction of a rule or a rulebase (as discussed in Section 2.3.3) from a set of observations, more formally axioms are constructed from the consequences of these axioms. The methods covered in this section are based on the induction of rules from data. The learning process employed within the FuzzyTL uses a Ad-Hoc Data Driven Learning (ADDL) approach. The ad-hoc method is based on a more generic Data Driven Learning (DDL) learning approach. DDL uses the structure of the data to form the basis of the learning parameters. It is prominent in dynamic environments as it is able to model varying forms of time-series data (Deshpande et al. 2004).

The concept of Data Driven Fuzzy Modelling (DDFM) can be placed within the wider scope of DDL. With its foundations in the seminal work of Zadeh (Zadeh 1965), fuzzy modelling has been adapted and implemented in varying domains from stock price analysis (Fazel Zarandi et al. 2009) to ecosystem management (Adriaenssens et al. 2004) and face detection (Moallem et al. 2011). Sugeno and Yasukawa have described the use of DDFM as a qualitative modelling approach. The qualitative nature of the modelling process allows for the representation of knowledge in a linguistic, humanistic manner, along with an ability to approximate non-linear models with simpler forms (Chen & Linkens 2001). Sugeno and Yasukawa (Sugeno & Yasukawa 1993) additionally define that fuzzy modelling has two aspects: structure identification and parameter identification. Structure identification can then be split into four sub-categories. The need to identify input candidates and input variables alongside the number of rules and the partition of the input space.

In the context of the FuzzyTL framework, to identify the structure and parameters of the FIS an automated learning process is used. Through the use of a numerical source data, fuzzy rules and fuzzy sets are formed using an ADDL method. The procedure is based on an algorithm proposed by Wang and Mendel (Wang & Mendel 1992). There are benefits for using this type of method in the extraction of a model from numerical data. Its simplicity makes it easily understandable and the nature of the low computation required allows for a greater speed of implementation. The swiftness in the execution within the early stages of the preliminary fuzzy modelling process allows for subsequent adaptation of the model by other methods (Casillas et al. 2000). Within the following sections, the WM method will be discussed in detail.

2.3.5.1 Wang-Mendel Methodology

The basis of the WM process is the formation of fuzzy sets and fuzzy rules that constitute the main components of the FIS. The main element of the method is the generation of fuzzy rules from numerical pairs which in turn are formed into a rulebase. The approach is a generalised one and is in keeping with the requirements of the transfer learning structure used in the FuzzyTL framework. Transfer Learning is discussed at length in Section 2.4. Wang and Mendel(Wang & Mendel 1992) proposed a four step procedure to produce the fuzzy rules and fuzzy sets.

Construct Fuzzy Regions Wang and Mendel's initial step is to divide each domain interval into fuzzy regions, each containing the membership functions for that input or output. Assuming that there are two inputs (x,y) and one output (z), the process is to divide each of these domains by $2N + 1$ regions where N can be different for differing variables. In order to automate this step, the domain is equally divided based upon the minimum and maximum values of the interval and the defined number of regions. Figure 2.5 shows the input and output domains divided into five regions and labelled with linguistic values VS (Very Small), S (Small), M (Medium), L (Large) and VL (Very Large).

The shape of the membership function can vary, however for this example triangular memberships functions were chosen.

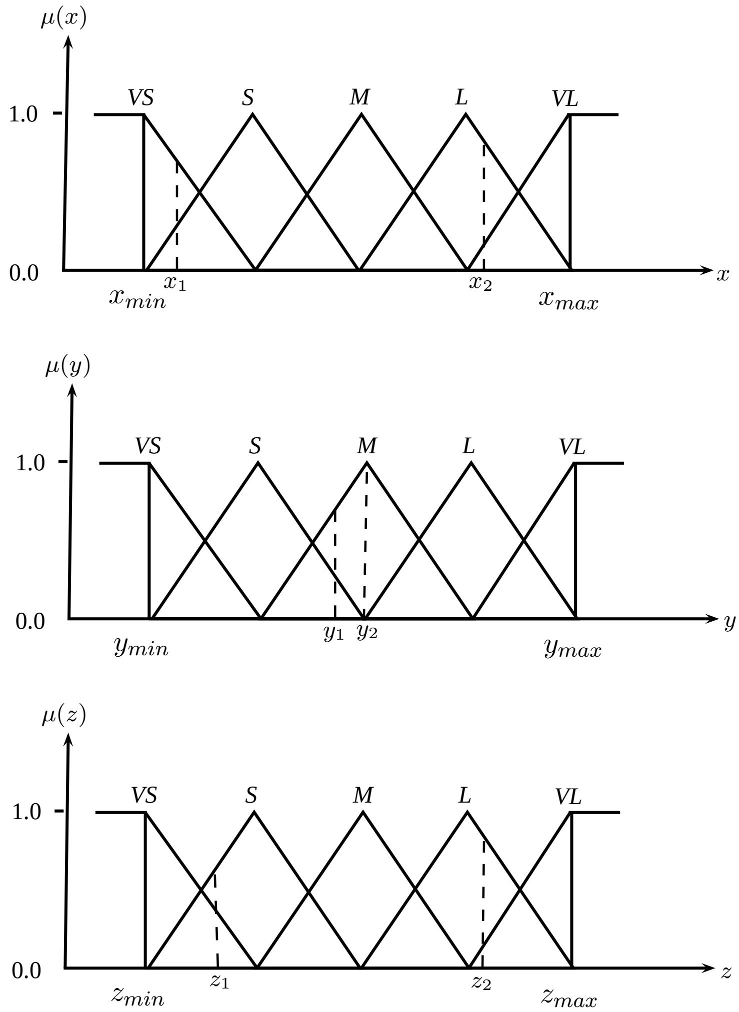

Figure. 2.5. Construction of Fuzzy Membership Functions Through the Use of the Wang-Mendel (Wang & Mendel 1992) Process.

Generate Fuzzy Rules To produce the fuzzy rules from the numerical data, the first step is to determine the degrees of membership from each data pair and generate an input-output rule. The maximum membership value of each input and output is taken of each individual data tuple. This can be demonstrated in the examples given in Figure 2.5. For input x_1 the membership values are 0.65 in VS and 0.35 in S with 0 in all other regions. The maximum is thus 0.65 in VS. Based upon this process the below linguistic rules can be produced.

$(x_1, y_1, z_1) \Rightarrow$ [x_1 (0.6 in VS), y_1 (0.7 in M), z_1 (0.55 in S)] $\Rightarrow$ Rule 1
IF x_1 is VS and y_1 is M **THEN** z_1 is S;
$(x_2, y_2, z_2) \Rightarrow$ [x_2 (0.75 in L), y_2 (1.0 in M), z_2 (0.75 in S)] $\Rightarrow$ Rule 2
IF x_2 is L and y_2 is M **THEN** z_2 is L;

Rule Base Reduction The production of the fuzzy rules can result in a rulebase that is equal in size that of the original dataset as each individual data point produces a single rule. This can become unmanageable in size. The construction of the rules from similar data points, can additionally result in conflicting elements. To reduce the rulebase size and remove conflicts, each of the rules are assigned a degree (d) based upon the maximum product of the individual inputs and outputs. The below equation depicts this:

d(Rule 1) = $max_{VS} \cdot (x) max_M \cdot (y) max_S(z)$
= 0.65 x 0.7 x 0.55 = 0.25025

Each rule is combined into groups based on the antecedent values. The rule with the highest degree in each group is kept within the rulebase. The other rules are removed. This produces the Reduced Rule Base (RRB). The WM approach also allows the option of the incorporation of expert knowledge in defining the rules that are used and/or kept.

Mapping of Output via Defuzzification The final stage of the process is to produce a mapping between the inputs and outputs. This is achieved by a defuzzification of the inputs. Wang and Mendel suggest a COG defuzzification strategy though there are a number of others that are applicable based on the context of the problem. This method produces an output value.

2.3.5.2 Other Rule Induction Methods

The autonomous extraction of fuzzy rules from data has produced a number of methodologies, implementing a number of differing strategies. The WM process (as illustrated in the previous sections) has been expanded and built upon. Sudkamp and Hamell (Sudkamp & Hammell III 1994) follow the method laid out by Wang and Mendel in (Wang & Mendel 1992) to construct fuzzy rules by dividing the input and output domains into regions. Sudkamp and Hamell extend the work of WM to introduce completeness. In this context completeness can be defined as a property of a rulebase that guarantees that there is at least one rule whose antecedent significantly matches every possible input (Sudkamp & Hammell III 1994, Lee 1990).

Learning from examples introduces the possibility that not all of the training data covers the whole of the feature space. As a result, certain inputs can result in no output being returned. A similarity and interpolation process is proposed to complete the rulebase.

Outside of the WM method there have are a number of methods that use the inductive approach to rule extraction. Ishibuchi et al. (Ishibuchi et al. 1994) have proposed a number of methods that use a grid partition to construct the sets. The most simplistic is the use of evenly divided domains to initially produce the fuzzy sets. Any membership function can be used although the most common is triangular. Taking all the possible combinations of the inputs, a set of rules are produced. This is a simplistic and computationally inexpensive method, however it can produce a large number of rules.

Nozaki et al. (Nozaki et al. 1997) take the simplistic division of the input space a little further with the incorporation of a simple heuristic to calculate the rule outputs. For rule i in an input domain divided evenly into K fuzzy sets, the heuristic states:

$$z = \frac{\sum_{j=1}^{n} w_i(j)y(j)}{\sum_{j=1}^{n} w_i(j)} \tag{2.9}$$

where z is the output, $y(j)$ is the data output and $w_i(j)$ is the i-th rule firing strength for the data pair.

Within this section a number of rule induction methods have been reviewed. The use of purely inductive methods are able to produce simplistic and effective fuzzy rulebases. These computationally inexpensive methodologies allow for further extension and adoption. Additionally methods such as WM are highly applicable to real-world settings where ease of implementation and low resource are necessary. This section focussed on the WM method as this is a major element of the FuzzyTL framework.

2.3.6 Fuzzy Clustering

Fuzzy clustering enhances the pattern recognition technique of cluster analysis to incorporate the uncertainty that can be described through the use of membership functions (Yang 1993). The methodology was originally proposed by Dunn (Dunn 1973) with further methods introduced by Bezdek (Bezdek 1973). For further insight see (Bezdek 1981, Bezdek et al. 1984).

A number of clustering methods to extract fuzzy rules have been proposed. Hong and Lee

(Hong & Lee 1996) propose a learning method to automatically derive fuzzy rules and membership functions from a set of data points. Training instances are combined with the use of an algorithm to provide a knowledge acquisition facility. The algorithm used by Hong and Lee (Hong & Lee 1996) is built around six steps:

Step 1: Cluster and fuzzify the output data.

Step 2: Construct the membership functions for the input variables.

Step 3: Cluster an initial decision table.

Step 4: Simplify the decision table.

Step 5: Adjust and reconstruct the membership functions in the simplification process.

Step 6: Produce decision rules from the decision table .

Again adopting a fuzzy clustering approach, Setnes (Setnes 2000) proposes a rule extraction method where each cluster corresponds to a fuzzy **IF-THEN** rule in the input-output product space. Setnes method uses Orthogonal Least Squares (OLS) to remove redundant or less important clusters during the clustering process. This process extracts the fuzzy rules that capture the important features of the systems input/output space. The result is a compact and transparent fuzzy rulebase.

Setnes and Roubos (Setnes & Roubos 2000) have approached rule extraction by combining fuzzy clustering with a Genetic Algorithm (GA). They applied a *c*-means clustering algorithm , first introduced by Bezdek (Bezdek 1981), to obtain a compact initial rule-base. The model was then optimised through the use of a GA.

The automated construction of fuzzy models from numerical data can result in redundancy in the form of similar fuzzy sets. Additionally, as the number of rules grows so does the complexity of the rulebase. Chen and Linkens (Chen & Linkens 2004) tackle these issues by using a simplification method based on both fuzzy clustering and optimisation. Their process implements partition validation combined with approximate similarity analysis. Adding to the methodology, they provide optimisation through a gradient-descent process.

The framework automatically determines the number of fuzzy rules from the fuzzy clustering procedure. Using simple equations for measuring the similarity of the fuzzy sets, the fuzzy structure is simplified by removing redundant sets and combining similar linguistic terms into a single linguistic value.

An overriding issue of fuzzy clustering stems from the need to define the quantity of clusters *a-priori*. Expert knowledge or previously induced models of the environment must be used to calculate the required number. The number of clusters is an important parameter as it has a direct impact on how the data is partitioned. If an incorrect number of clusters are chosen, misclassification can occur as the clusters are not well separated and compact.

2.3.7 Neural Networks

A neural network, or Artificial Neural Network (ANN), is a learning process inspired by aspects of the human brain. An ANN can broadly be defined as:

> "an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the interunit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns" (Anderson & Davis 1995).

There has been considerable development in introducing the use of Artificial Neural Networks (ANNs) to assist in the construction of fuzzy systems. The use of Neuro-fuzzy systems have been applied to varying domains including control, data analysis and decision support (Nauck 1997). A neuro-fuzzy system can be defined as:

- A fuzzy system that is trained by a learning algorithm derived from a ANN.
- The units in the network are t-norms or t-conorms rather than activation functions.
- The system can be interpreted as a set of rules.
- The semantic properties of the fuzzy system are taken into account (Nauck 1997).

Kasabov (Kasabov 1996) introduced a methodology for the use of neural networks for the learning of fuzzy rules based on a Fuzzy Neural Network (FNN) approach. The methodology can be summarised as having two major aspects. Firstly, a set of fuzzy rules are used to define the initial structure of the Neural Network (NN) which is trained on a set of data. Secondly, after the structure has been defined, parameters are observed that are used to derive the rulebase. The rules are represented linguistically. The model presented by Kasabov uses an Multi Layer Perceptron (MLP) and a backpropagation training algorithm. The model used is adaptive across both the membership functions and the fuzzy rules.

There are a number of issues with the production of fuzzy rules using ANNs, many based on the need for *a priori* knowledge. Wu et al. (Wu et al. 2001) instigated the use of ANNs to acquire fuzzy rules in an automated fashion without the need for previous knowledge. The system is initialised without a rulebase. Rules are added or removed dynamically in response to the level of their significance to the systems performance. This allows for the structure to be self-adaptive.

The system employs a lack of iterative learning so the learning speed is very fast. Through the use of pruning, significant nodes are selected so that a frugal structure with high performance can be reached. Wu et al. (Wu et al. 2001) identified a number of issues with the production of the fuzzy rules.

- Widths of the membership functions are the same due to the nature of the ANN used.
- The number of membership functions is the same as the fuzzy rules.
- A number of parameters need to be initialised randomly which can result in large set widths and difficult user implementation.

Leng et al. (Leng et al. 2005) also approach the problem of extracting fuzzy rules through the use of a hybrid ANN, namely the implementation of a Self Organising Neural Network (SONN). The basis of the work is to produce a self-organising neural network (SOFNN) that implements a Takagi-Sugeno fuzzy system on-line. The algorithm that is proposed can be divided into two elements: parameter learning and structure learning. The parameter learning aspect of the system uses a modified recursive least squares approach. The structure learning is derived from geometric growing (Leng et al. 2005) and applies a pruning method based on the optimal brain surgeon approach with proposed parameter learning.

2.3.8 Evolutionary Computation

Evolutionary Computation (EC) is a broad subfield of CI that incorporates a range of optimisation techniques inspired by evolutionary mechanisms, primarily using evolutionary algorithms. EC has its foundations in four evolutionary approaches: Evolutionary Programming (EP), Evolution Strategies (ES), GA and Genetic Programming (GP). For a further insight into these methods see (Fleming & Purshouse 2002).

Casillas et al. (Casillas et al. 2000) used a GA to adapt the WM method. A GA takes the inspiration for its learning process from evolution. They are global optimisation techniques that strive to remove some of the failures of local searches. A GA is an iterative search which produces and maintains a population of candidate solutions. Throughout each iterative step, referred to as a generation, the population is evaluated based on its structure. Dependent on the outcome of the evaluation, a new population of solutions is formed. The initial population can be chosen heuristically or at random. Some variation is introduced so that all areas of the feature space are searched (Grefenstette 1986). Castillas et al's approach was to use the ADDL technique of WM and incorporate cooperative rules. The proposed methodology performs a search in the set of candidate rules produced by the WM process (see Section 2.3.5.1 for more details). Within the candidate set, the optimisation technique strives to obtain the best joint accuracy across the fuzzy input space through the use of the cooperation.

Using a similar approach Ishibuchi and Yamamoto (Ishibuchi & Yamamoto 2004) use a GA to construct a fuzzy rulebase. The method used is to select a small number of fuzzy rules from a larger candidate selection. The production of the small rulebase is initialised through the use of a Multi-Objective Genetic Algorithm (MOGA) which is extended to a Multi-Objective Genetic Local Search (MOGLS).

The first process in the algorithm is to pre-screen the candidate rules. The basis of this stage is the production of a *confidence* and a *support* value. The calculation of these values are based on measures used in evaluating association rules. By focussing on the number of training patterns within the data set that are compatible, the values can be produced. Using a combination of the consequent and pre-screening values, groups are produced and placed into descending order. A section of these rules are then chosen based on a user defined measure. This reduces the number of rules that need to be processed. It is impractical to examine all combinations of rules when there is a large number of input variables (Ishibuchi & Yamamoto 2004).

The second stage of the process is the use of the MOGLS. The MOGLS algorithm is implemented with rule weight learning. Overall, the use of both pre-screening and the MOGLS can result in a compact rule set with high interpretability.

Although very popular, well established and mature, other evolutionary methods have been used outside of GA's. Cabrita et al. (Cabrita et al. 2006) have implemented fuzzy rule extraction through the use of a memetic approach. By using a hybrid of an NN and a memetic approach, Cabrita et al. present a Bacterial Mememtic Algorithm (BMA). Through the mimicking of microbial evolution and gene transfer, an optimal fuzzy rulebase is produced to classify a pattern set. The basis of the algorithm is to encode a randomly created population of fuzzy rules into a population of chromosomes. Following bacterial mutation and the application of the *Levenberg-Marquardt* method, the rules are evaluated against a criteria until an optimal solution is reached (Gal et al. 2008).

Yang et al. (Yang et al. 2010) also propose the use of a EC method to extract fuzzy rules. A metaheuristic that has ties to evolutionary programming, Particle Swarm Optimisation (PSO) is a technique that represents the movement of flocks, herds or schools of creatures (Kennedy & Eberhart 1995). Yang et al. proposed the use of a PSO to improve the steps that Wang proposed in (Wang 2003) by optimising the fuzzy rule centroid of the data covered area to improve forecasting accuracy.

2.3.9 Online and Evolving Fuzzy Systems

Evolving Fuzzy Systems (EFS) are a branch of FL that have emerged in recent years. As with many fuzzy systems, EFS are based on FIS's, however they are self-developing, and self-learning. The principles of the EFS methodology looks to address the changing environmental conditions of real-world problems (Angelov & Buswell 2001). Angelov and Zhou (Angelov & Zhou 2006) comment that evolving fuzzy systems mimic the evolution of individuals in nature. The methodology mirrors the developmental processes associated with learning from experience and inheritance. They draw an analogy with the way in which people learn. An individual will start with an empty rulebase. New rules are added from life experience based on data streams. The development of the rules are gradual. The rulebase itself is not fixed or pre-defined. EFS has been applied to a number of areas from intelligent sensors (Angelov & Kordon 2010) and health monitoring (Filev & Tseng 2006) to robotic applications (Zhou & Angel 2006).

EFS have many attributes in common with the use of online learning systems. Angelov and Buswell (Angelov & Buswell 2001) state that the EFS is an online approach to the adaptation of the fuzzy rulebase. In a similar fashion to EFS, online learning and particularly online fuzzy systems adapt their construction based on information gained from the target. Using the definition put forward by Hagras et al (Hagras et al. 2003, Hagras et al. 2004), any learning carried out with user intervention and in isolation from the environment using simulation is defined as *offline learning*. In cases were the learning has interaction with the actual environment, this is referred to as *online learning*. Online learners differ from standard learning mechanisms in the way in which new hypotheses are constructed. Casa-Bianchi et al. (Cesa-Bianchi et al. 2004) state that online learners feed in a hypothesis $h \in \mathcal{H}$ and an example data point (x, y), and return a new hypothesis $h^j \in \mathcal{H}$. Based on a set of data points $Z^n = ((x_1, y_1), \ldots, (x_n, y_n))$, a set of (not unrelated) hypotheses will be created.

The methodology outlined in Chapter 3 incorporates the use of an online learning method incorporating the adaptation of both fuzzy sets and fuzzy rules, reflecting changes in the target task.

2.3.10 Discussion

Within this section an overview of the FIS was given with an introduction to FL, fuzzy rules and fuzzy sets. Combined with this, a discussion was provided into varying methods of producing fuzzy rules in an automated fashion. Particular emphasis was given to the WM methodology as this is the approach used within this thesis.

A number of attributes can be afforded to WM methodology which are suitable for FuzzyTL. These can be summarised as:

- Ease of implementation.
- Ability to modify the framework.
- Mature and well established.
- Implemented across a wide number of applications.

- High generality.

The simplistic and easy approach to the implementation of the WM method allows for a quick adoption of the framework. Equally, the ability to modify its components are fundamental to the FuzzyTL frameworks adaptive approach. Its level of generality and maturity have allowed it to be applied to a large number of applications. This has been demonstrated with the methods capacity to be implemented in a broad number of applications.

When solving the problem set out within this thesis, the learning methods discussed encounter specific issues. The nature of the fuzzy clustering process requires that the cluster quantity is defined *a priori*. This may not always be possible. The search for optimal cluster numbers is a continuing area of research. Along with the number of clusters, the cluster centroids and location are also not known *a priori*. As a result initial estimates are needed. The clusters themselves can also produce large variability. The quantity of data points, the density and the variability can cause classification issues (Gath & Geva 1989).

Similarly within the use of a ANNs, the initialisation of the learning process can produce the need for *a priori* information. A specification must be made of each input variable fuzzy partition through the initial fuzzy sets (Nauck 1997). An additional problem with neuro-fuzzy systems is the role of rule learning. Many systems have no rule learning defined or only use simplistic heuristics. These simple approaches, however, are not sufficient enough to produce small and interpretable fuzzy rulebases. In many cases rule pruning to reduce the fuzzy rulebase, and fuzzy clustering methods to find fuzzy rules are proposed.

The use of a GA also can provide certain difficulties. They can be computationally heavy weight, with large data sets often needing extensive processing due to nature of the fitness function evaluation. Although the near-optimal solution can be found, there is a need to understand the nature of parameters to achieve the best outcome (Casillas et al. 2000). Some processes have shown that although a reduction in the domain will bring more efficient results, some knowledge is necessary to calculate the parameters to reduce the initial search space.

2.4 Transfer Learning

Transferring learning is a humanistic trait that has been well studied across education, psychology and philosophy (Perkins & Salomon 1992, Macaulay 2001). In education, Transfer Learning (TL) or the transfer of learning is referred to as

> "prior-learned knowledge or skills that affect the way in which new knowledge or skills are learned and performed. Transfer is deemed to be positive if acquisition and performance are facilitated, and negative if they are impeded" (Leberman et al. 2006, McKeough et al. 1995, Cormier & Hagman 1987).

When applied to a CI domain, the goal of transfer learning is similar. The motivation of transfer learning is to improve the learning in a target domain by acquiring information from a differing but related domain. Traditional machine learning strategies work under a number of assumptions. Mihalkova et al. (Mihalkova et al. 2007) propose that the learning of each new task begins from scratch. Additionally, there is a need within the majority of machine learning techniques that the data used for training and testing is required to come from the same feature space. TL offers the ability to use previously acquired knowledge to improve the learning in a related area. TL can be applied to varying domains. As an example, a web documentation task has been undertaken to manually label web site documents into defined categories. As a new website is created, the data features and data distributions are different to those contained within the old site. There is a lack of training data to categorise the new pages. TL can transfer the classification knowledge to the new domain (Dai et al. 2007). A major motivation behind the FuzzyTL framework comes from environments that lack any prior knowledge in the form of labelled training data. TL is incorporated into the novel methodology presented in this thesis to address issues that arise from the lack of available training data.

Within this section, a definition of TL is given that will be used in this thesis. As the FuzzyTL framework incorporates both TL and FL, a discussion is subsequently given on differing CI techniques that have been used with TL.

2.4.1 Measures, Definition and Foundations

Transfer learning contains two principle elements, a *Domain* and a *Task*. According to Pan and Yang (Pan & Yang 2009), a *Domain* can be defined as consisting of two components: a feature space x and a marginal probability distribution $P(x)$ where $X = \{x_1, \ldots, x_n\} \in X$. A *Task* consists of a label space Y and a predictive function $f(.)$. The predictive function can be learned from the training data which is constructed as data pairs $\{x_i, y_i\}$ where $x_i \in X$ and $y_i \in Y$. The *source domain* can be defined as $\mathcal{D}_s = \{(x_{s_1}, y_{s_1}), \ldots, (x_{s_n}, y_{s_n})\}$ where $x_s \in X_s$ is the data point and $y_s \in Y_s$ is the corresponding label. The *task domain* can be defined as $\mathcal{D}_t = \{(x_{t_1}, y_{t_1}), \ldots, (x_{t_n}, y_{t_n})\}$ where $x_t \in X_t$ is the data point and $y_t \in Y_t$ is the corresponding output.

Based on these definitions transfer learning can be defined as:

> Given a source domain $\mathcal{D}_s$ and a learning task $\mathcal{T}_s$, a target domain $\mathcal{D}_t$ and a learning task $\mathcal{T}_t$, transfer learning aims to improve the learning of a new task $\mathcal{T}_t$ through the transfer of knowledge from a related task $\mathcal{T}_s$ (Torrey & Shavlik 2009) by the learning of the predictive function in the target domain $\mathcal{D}_t$, where $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$ (Pan & Yang 2009).

Torrey and Shavlik (Torrey & Shavlik 2009) set out three measures using this formal definition of transfer learning to monitor the possible improvements through the use of transfer learning.

1. The initial performance achievable in the target task using only the transferred knowledge, before any further learning is done, compared to the initial performance of an ignorant agent.

2. The amount of time it takes to fully learn the target task given the transferred knowledge compared to the amount of time to learn it from scratch.

3. The final performance level achievable in the target task compared to the final level without transfer (Torrey & Shavlik 2009).

Using these measures it is possible to calculate the improvement through the implementation of a transfer learning scheme. An adaptation of these measures inform the experimentation criteria used in Chapter 4.

2.4.2 Background

The beginnings of TL stem from a number of areas, however, it is recognised that Multi-Task Learning (MTL) was highly influential. MTL is a learning methodology that uses parallelisation of tasks to learn whilst sharing information contained within the domain. Caruana (1997) describes an example. Four independent tasks are processed using four ANN working in isolation to produce a value. This is classed as Single Task Learning (STL). By combining the tasks as inputs into a single ANN and sharing a common hidden layer, the internal representations that are produced for one task can be shared. This single backpropagation NN would subsequently produce four separate outputs. A fundamental concept of Multitask learning is the sharing of previously learnt information from different tasks while they are trained.

Work by Thrun and Mitchell (Thrun & Mitchell 1995) also paved the way to establishing Transfer Learning. They proposed a methodology that uses task-independent knowledge learnt over the lifetime of a robot's activities. The methodology generalised control tasks and subsequently reduced the need for further experimentation. Thrun and Mitchell highlight four areas were limitations occur due to the complexity of the environments that they operated within:

Knowledge Bottlekneck: A human designer is limited in their capacity to provide an accurate model of the world and the robot.

Engineering Bottlekneck: The supply of sufficiently detailed knowledge in a computer accessible form can be complex and extensive.

Tractability Bottlekneck: Some robot domains are too complex to handle efficiently.

Precision Bottlekneck: Difficulty can arise in producing robots accurate enough to execute plans that are generated using the internal models of the world.

There are clear comparisons between the Lifelong Learning methodology and a basic humanistic form of learning. Each can be defined by related control tasks that are encountered over a continuous period of time. When faced with a new task to learn, humans are usually able to call upon information formed from previous experiences. These will stem from other, related learning tasks (Thrun 1996). Thrun and Mitchell go on to define the concept of Lifelong Learning and its necessity in the context of robotic control, and make this comparison. In a similar vein to Multitask Learning, lifelong learning is proposed to reduce the difficulty encountered in solving a related control problem by using knowledge that has been acquired from solving earlier control problems. There is additional discussion of the feature space when defining the control problem. If the robot remains the same, the sensors and effectors will equally. However, the environment and the task (in this case reward function) may change. This is an important part of the knowledge transfer process.

2.4.3 Transfer Learning Types and Variations

In the following section an overview of the different types of TL will be given with a discussion of the applicability of each learning strategy.

2.4.3.1 Unsupervised Transfer Learning

As with other forms of TL, unsupervised TL looks to improve the predictive function in the target domain by extracting information from the source to assist the target. Taking a similar stance to standard unsupervised learning, the data within neither the target or source domain contains labels.

Cook et al. (Cook et al. 2012) deviate from the standard terms of *supervised* and *unsupervised* learning. They introduce the use of *informed* and *uninformed* which are applied to the availability of labelled data in the source and target areas. Informed Supervised (IS) transfer learning implies that labelled data is available in both the target and source domains. Informed Unsupervised (IU) transfer learning, however defines that the labelled data is only available in the source domain. By contrast, Uninformed Supervised (US) learning implies that labelled data is available only in the target domain with Uninformed Unsupervised (UU) transfer learning implying that there is no availability of labelled data in either domains.

Different methods have been employed to achieve results in this area though they are dependent on similar restrictions to those exerted by standard unsupervised learning approaches. The work by Raina et al. (Raina et al. 2007) falls into the category of unsupervised transfer learning. They developed a method defined as *Self-Taught Learning*. Within Self-Taught learning, unlabelled data is used in a supervised fashion. Classification of images is improved by using a large dataset of images from the internet/Web resources. This is combined with a sparse coding approach to construct high level features.

2.4.3.2 Inductive Transfer Learning

Inductive transfer learning is derived from classical inductive learning. The target learning task is different from the source learning task. Labelled data in the target domain is required to induce an objective predictive model. In inductive transfer learning both $\mathcal{D}_s = \{(x_s, y_s)\}$ and $\mathcal{D}_t = \{(x_t, y_t)\}$ are known. Additionally there is auxiliary unlabelled data that is not part of the training set (Pan & Yang 2009). The way in which the target task is altered by the source task knowledge is based on the specific inductive learning algorithm used (Torrey & Shavlik 2009).

Cook et al. (2012) propose that there needs to be a more complete taxonomy of inductive *and* transductive learning when referencing transfer learning. They specify that inductive learning requires that labelled data is available within the target domain, whether or not it is available in the source (Cook et al. 2012). As a result most supervised and IU transfer learning techniques are inductive.

2.4.3.3 Transductive Transfer Learning

Transductive transfer learning requires that the source and target learning tasks be the same, but the domains may differ (Arnold et al. 2007). Pan and Yang (2009) further define that a quantity (not all) of the unlabelled target data is required during training to produce a marginal probability for the target data. Within the context of transfer learning, transductive deviates from the standard machine learning meaning. In this area of research it predominantly refers to the tasks being the same and unlabelled data being available in the target domain. Within Cook et al's definition (Cook et al. 2012), uninformed supervised methods are additionally transductive TL techniques.

2.4.3.4 Negative Learning

Negative transfer and negative learning has parallels with human learning. Perkins and Salomon (Perkins & Salomon 1992) comment that within education

> "negative transfer occurs when learning in one context impacts negatively on the performance in another".

Any TL method strives to improve the learning process of the target domain. The effectiveness of the transfer method depends on the relatedness of the source and target domains (Thrun 1996). The overall goal of the TL method is to increase the performance of the method whilst avoiding negative impact. This can be a difficult statement to realise. The approach of the transfer method and its relative caution, can have a direct relationship to the positive learning that occurs. Methods that have safeguards against negative transfer often produce fewer positive increases whilst aggressive strategies with no protection produce larger positive transfer.

There are a number of methods that have approached negative learning. These can be summarised within three areas:

Incorrect Information The transfer approach attempts to recognise and reject harmful source knowledge while learning the target task. This approach can remove the source completely so that the learning is no worse than if the target had no extra information. Luo et al. (Luo et al. 2012) use a Active Vector Rotation (AVR) to select a small set of data points from the source to initialise the learning process within the task. The instances are weighted from the source so the possibility of negative impact is reduced.

Selection of Source Task If there exists the option of more than one source task, the need for the TL algorithm is to acquire the most fitting task. Negative transfer can be reduced despite algorithms having little protection by selecting the best source. Talvitie and Singh (Talvitie & Singh 2007) map a target task to a related task based on the tasks current situation. A sequential decision making process is represented as a Markov Decision Process (MDP). An agent has a group of candidate policies which are generated from the source and the target task. A decision is made on an optimal policy to use for the target. Talvitie and Singh use the analogy of a group of experts offering advice. The agent must leverage their knowledge to learn a solution. The agent can also ignore the advice and learn the task from scratch.

Task Similarity To reduce the risk of negative transfer, some approaches model the relationship between the tasks. The basis of the similarity can lead to a better use of the source information. Cao et al. (Cao et al. 2010) use an automatically learned transfer scheme to produce a transfer kernel. The transfer kernel models the correlation between the tasks to produce a measure of similarity. The transfer is then based on how similar the source is to the target task.

2.4.3.5 Limited-Data Transfer Learning Methods

Within the transfer learning framework there are a number of sparse and limited-data methods. In this context limited-data refers to scenarios with datasets that are a low percentage of the overall quantity. This can be as low as a single data point. These lend themselves to real-world applications where little data is often available. One of these approaches is One-Shot Learning. The basis of One-Shot Learning can be identified by drawing parallels with the abilities of humans to identify objects under a wide variety of conditions after seeing only a single point. Miller (Miller 2002) sets out a one-shot learning approach using a transfer basis. The process acquires knowledge from one setting and uses it in another. The methodology models new classes of objects based only on samples of related or support classes. One-shot Learning is carried out on object classes that are variable such as hand written characters or writing whose lighting conditions have changed. Probability densities are developed over common image changes to form a model. A combination of a generic model of image change is used with a single sample of a new object to provide the new model. This model is then used for synthesis, classification and other visual tasks.

Larochelle et al. (Larochelle et al. 2008) expand the concept of limited data with the introduction of *Zero-data Learning*. Zero-data learning is based on the premise that a model must generalise to classes or tasks where there is no availability of training data, only a description of the data. It is assumed that the situation may occur that no labelled data is available so descriptions are used. There are similarities between the problem proposed within this thesis and that approached by Larochelle et al. Larochelle et al. assumed that the descriptions that are used within the classification process are predefined. Within *Zero-data Learning* the hierarchical definitions can often come from expert opinion, differing from the automated, data-driven strategy chosen within this thesis.

2.4.3.6 Transfer Learning With Computational Intelligence

To construct the algorithms that constitute the learning methods within transfer learning, a number of differing CI methods have been employed. Within this section, some of the major variations will be discussed.

Transfer Learning Using Genetic Algorithms: A goal of transfer learning is to increase the speed of the learning process by incorporating differing, but related task data. Taylor et al. (Taylor et al. 2006) introduce the use of GA's and TL by extending a previously constructed algorithm to endeavour to achieve this goal. Their approach is to extend the TL method of producing a translation function. This process allows for differing value functions that have been learnt to be mapped from source to target tasks. Taylor et al. (Taylor et al. 2006) incorporate the use of a set of policies originally constructed by a GA to form the initial population for training the target task. They show that transfer of inter-task mappings can reduce the time required to learn a second, more complex task.

Transfer Learning Using Neural Networks: In the scope of transfer learning, Collobert and Weston (Collobert & Weston 2008) apply the use of a deep ANN architecture for Natural Language Processing (NLP). They use a ANN architecture that when given a single sentence will output a host of predictions: tags, named entities, semantic roles, semantically similar words and the likelihood that the sentence makes sense. Feature extraction is placed across the sentence on a number of layers. The features in deep layers of the ANN are trained automatically to the relevant task. All the tasks within the feature extraction are jointly trained with the exception of the language model. The training of this model is carried out through semi-supervised and multi-task learning.

Transfer Learning Using Reinforcement Learning: The concept of Reinforcement Learning (RL) is based on trial and error (Sutton & Barto 1998), a way of programming an agent through reward and punishment without the need to specify how the task is completed (Kaelbling et al. 1996). Figure 2.6 shows a standard reinforcement model.

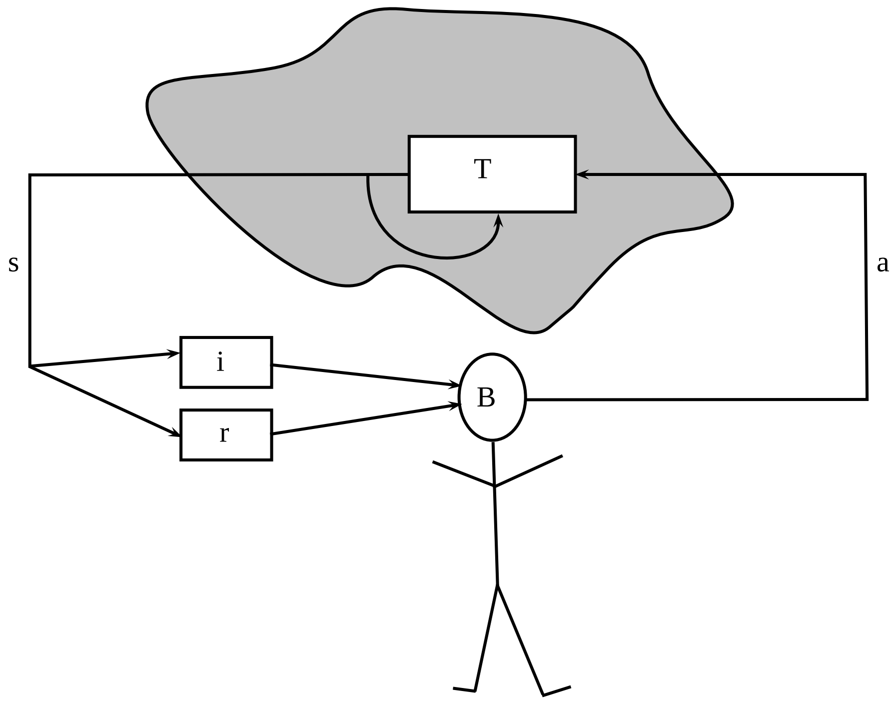

Figure. 2.6. The Standard Reinforcement Model Adapted From (Kaelbling et al. 1996).

An agent, depicted in the diagram as a person, is connected to its environment T via perception and action. At each step of the interaction, the agent receives inputs (an indication of the current state i and the current state of the environment s). The agent uses these to generate an action output, a. The action changes the state of the environment which the state transition is communicated to the agent through a scalar reinforcement signal r (Kaelbling et al. 1996). The behaviour of the agent B should choose actions that increase the sum of values of r. This process can be learnt through trial and error with additional guidance from learning algorithms.

Barrett et al. (Barrett et al. 2010) implement a RL based transfer learning method in physically grounded robots. The robots are trained in a controlled environment in order to deal with expected situations. As the robot encounters unexpected events, the process of RL is used to learn behaviour. As the learning of behaviour can be costly, the reuse of prior information is used in order to increase the speed of the learning.

Transfer Learning using Dimensionality Reduction: High dimensional datasets present mathematical challenges. Not all of the variables that are present in a dataset are necessarily required to understand the context that is being studied (Fodor 2002). Dimensionality reduction attempts to address this problem by reducing the amount of variables under consideration. The problem can be formally described as follows: given a p-dimensional random variable $X = (x_a, \ldots, x_p)^T$, the goal is to find a lower dimensional representation of it where $s = (s_1, \ldots, s_k)^T$ and $k \leq p$ whilst capturing the content in the original data according to a defined criteria.

Pan et al. (Pan et al. 2008) propose the exploitation of the latent space that exists between source and target domains as a bridge to facilitate knowledge transfer. The methodology is based upon finding close marginal distributions between the source and target data. Specifically, if the two domains are related, there can exist common latent variables that are contained within the observed data. A portion of the variables may have a negative impact on the distributions of the observations. Equally, others can have a positive impact. Taking the latent factors that do not cause change across the domains, a lower-dimensional space is formed.

2.4.4 Other Learning Methods

By its very nature TL is an overarching methodology taking influence and encapsulating other processes. In this section, related learning methods will be discussed.

2.4.4.1 Semi-Supervised Learning

Semi-Supervised Learning (SSL) sits in between supervised and unsupervised learning. Unlike supervised learning, where the goal is to learn a mapping from x to y, given a training set of pairs (x_i, y_i), SSL is supplied with unlabelled data. $y_i \in Y$ is referred to as the labels of x_i. Typically the focus of unsupervised learning is to find structure in the unlabelled data, $X = (x_1, \ldots, x_n)$ where $x_i \in X$ for all of $i \in n$ (Chapelle et al. 2006). SSL can be defined as a learning method that uses a quantity of unlabelled data, together with labelled data to build more efficient and cohesive classifiers (Zhu 2006). The data $X = (x_i)$ $i \in n$ can be divided in two segments. The first $X_l = (x_1, \ldots, x_l)$ for which labels are provided, $Y_l = (Y_1, \ldots, Y_l)$ and unlabelled points $X_u = (x_{l+1}, \ldots, x_{l+u})$. SSL predominantly focuses on the classification problem space (Zhu et al. 2003, Erkan et al. 2007). A distinction can be found between semi-supervised learning and transfer learning. Semi-supervised learning assumes, in the most part, that the data comes from the same data distribution. However, TL allows for the domains, tasks and distributions to be different (Pan & Yang 2009).

Liu et al discuss the use of a dynamic fuzzy semi-supervised multitask process. This framework is based on the use of a semi-supervised multitask learning process combined into a single framework (Liu et al. 2009). The authors expand semi-supervised fuzzy pattern matching to use attributes from differing sources to classify the target domain.

2.4.4.2 Domain Adaptation

Domain adaptation has many similarities to transfer learning in form and approach. Largely based on statistical classification, domain adaptation focusses on the basic assumption that although training and test data used for many learning methods come from the same distribution, the application data does not. Domain adaptation focuses on the use of *in-domain* data that is related to but not within the same distribution as *out-of-domain* data (Daumé III & Marcu 2006). This is in contrast to the problem of multi-task learning (Caruana 1997) where the distribution of the data does not change, while the task can vary from source to target. There is also a strong relationship between domain adaptation and semi-supervised learning (see Section 2.4.4.1). It can be considered that domain adaptation sits within the wider area of TL. Based on this TL can be considered not to be a defined methodology, however an overarching architecture. For this reason, domain adaptation methods will be viewed as being inclusive of transfer learning.

2.4.5 Discussion

In this section a broad outline of transfer learning was given with particular focus on supplying:

- A definition of transfer learning.
- An overview of the types of transfer learning.
- An overview of applications of computational intelligence within transfer learning.
- A broad look at comparative learning methods linked to transfer learning.

The preliminary parts of this section set out to offer the reader the background knowledge needed to understand TL. As discussed, TL forms one of the main elements within the FuzzyTL framework. The methodology approaches the issues that arise through the lack of available training data by incorporating knowledge from contextually differing, but similar implementations.

Additionally within this section, by illustrating the applications of CI, it was shown that the overarching framework of TL has the ability to incorporate additional learning methodologies. A full explanation of the incorporation of FL and TL within the FuzzyTL framework will be given within Chapter 3.

2.5 Intelligent Environments

IEs can produce data that contains both uncertainty and vagueness. There has been much research into modelling the uncertainty of IEs with FL techniques. This section discusses work and applications of FL to IEs. The initial section will focus on an overview of IEs with particular scrutiny of the varying definitions used within the area. A background is given of the varying definitions that surround Smart Environment (SE), Ambient Intelligence (AmI) and IEs. This understanding of terminology and definitions that will be used within subsequent sections of this thesis.

2.5.1 Definition

The concept of integrating people, devices and computation that was constructed by Weiser (Weiser 1991) in the early 1990's laid the foundations for what is referred to as pervasive or ubiquitous computing. Intelligent, Smart and AmI environments stem from this foundation. Pervasive Computing (PerCom) is

> "based on the integration between computer processing and common-use objects by means of small micro systems whose presence cannot detect or are not interested to detect" (Genco & Sorce 2010).

A pervasive computing environment can also be characterised as

> "one saturated with computing and communication capability, yet so gracefully integrated with users that it becomes a *technology that disappears*" (Satyanarayanan 2001).

Satanarayanan goes on to state that pervasive computing environments consist of four elements. The technology must firstly disappear from the perception of the user. The goal of the system is for interaction between user and the technology at an almost subconscious level. Secondly, the scalability of the implemented system is primary to its usability. For example, the quantity of users should not distract from how the system responds. As further mobility and complexity occurs, the implementation should absorb this. The third element is the masking of uneven environments. As a user moves from office to office, workplace to home, their perceptions of the technology that is around them should be managed. How smart an environment is perceived to be is linked directly to the perception of invisibility. The fourth element relates to what Satanarayanan describes as the effective use of Smart Spaces. This is the bringing together of systems, such as sensor technology, to intelligently control the environment of homes, or software applications, that alter behaviour according to a users location. This element relates directly to the concept of intelligent, smart and ambient IEs, and how they are defined, however, it is a departure from the more overarching and abstracted view previously discussed.

The definitions IE, SE and AmI are often used indiscriminately within literature and have a direct relationship to the concept of PerCom. There are a number of varying definitions for each concept. Cook and Das in reference to Youngblood et al. (Youngblood et al. 2005) state that a smart environment is

> "one that is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment" (Cook & Das 2007).

Cook and Das expand on this by focussing on the predictive and decision making requirements of an SE. To improve the experience of the user in the environment there is a need for the software within the SE system to be fully automated and adaptive, so removing the control that lies with the user. This requires the software to improve its performance over time through knowledge acquisition (Cook & Das 2004). Although more in tune with Satanarayanan's definition of Pervasive Computing, the incorporation of intelligent software that has a decision making capacity alongside an adaptive nature is more focussed than the general view of PerCom.

Similarly AmI and Ambient Intelligent Environment (AIE) are categorised in the same fashion to SE's. Hagras et al. (Hagras 2007) define AIE's as relying on ubiquitous computing technologies to implement the hardware structure that they need to operate. However, AIEs require a distributed intelligence such as intelligent agents to create a pervasive layer of intelligence within the system (Hagras 2007). This definition runs in parallel to that of an SE but again emphasises differences between AIEs and PerCom. The relationship between PerCom and IEs, SEs and AmI produces a hierarchical structure. Whilst the perceptual nature of Pervasive Computing separates it from computing in general, IEs can be viewed as a prerequisite to pervasive computing (Saha & Mukherjee 2003). Overall pervasive computing encapsulates IEs within a higher level, more generalised definition. Although sharing a high proportion of attributes, the focus of IEs on context-awareness, intelligent control and a *use* of a pervasive computing system differentiates the two concepts.

To summarise, IEs, SE and AIEs define the same concept. Adapting the definition of Cook and Das, these can be broadly represented as:

> "A system to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment through the use

of an intelligent structure that demonstrates adaptive, predictive and decision making capabilities" (Cook & Das 2007).

Within this thesis IE, SE and AmI will be used interchangeably and will refer to the definition outlined.

2.5.2 Computational Intelligence in Intelligent Environments

Computational Intelligence covers a wide breadth of techniques and processes. Many of these have been successfully used within IEs to model, adapt and optimise applications produced for IEs. Within this section, the use of FL techniques (see section 2.3 for an explanation of FL) will be discussed within the scope of AmI, smart and IEs.

IEs exemplify the dynamic nature of real-world applications that can produce vagueness and imprecision. As a result of this, FL has been used to assist in the modelling process, the learning of models and with decision making structures of many IE implementations.

Combining context-aware and a fuzzy approach, the work of Copetti et al. (Copetti et al. 2009) incorporates a reasoning module based upon the use of FL rules. The reasoning module forms a part of the Health Support in Aware and Ubiquitous Domestic Environments (H-SAUDE) framework. As a major element of the framework, the reasoning module is the basis for the decision making function. The module receives preprocessed data from sensors as inputs and subsequently conducts analysis to determine critical and emergency situations relating to an individuals hypertensive condition. The system is based upon a process of FL modelling created using an expert system. The initial stage, as with all FIS is the formation of fuzzy rules. This is achieved through the use of medical knowledge relating to the monitoring of key attributes. Copettis et al gives an example of this type of rule as:

> **IF** the average systolic pressure is greater than 135mmHg **AND** the diastolic is greater then 85mmHg **THEN** the patient is considered hypertensive (Copetti et al. 2009).

The decision making necessary to produce a valid output is formed from the use of fuzzy sets associated with both the medical diagnostics and the patients behaviour. The system outputs a value relating to the patients state (normal, alert or emergency). A historical analysis of the patients information is produced to individually assess the context over time. For example, if a patients status update occurs infrequently, the output is moved to an emergency level. This generated data is stored to help the next decision.

2.5.2.1 Multi-Agent Adaptive Fuzzy Systems

Multi-Agent: Doctor et al. (Doctor et al. 2005) incorporated a fuzzy learning and adaptation technique within an AmI environment Intelligent Dormitory (iDorm), also known as iSpace in

Essex, UK. Doctor et al. (2005) developed a life-long learning method using multiple intelligent agents incorporated into an IE environment. The life-long learning structure is based upon a Fuzzy Logic Controller (FLC) that uses a model free approach. The technique is referred to as Adaptive Online Fuzzy Inference System (AOFIS). This is based upon a unsupervised data-driven one-pass approach for extracting fuzzy rules and membership functions from data. The FLC is used to model the users behaviour within the IE.

To produce the FLC, Doctor et al proposed that the data is gathered by the monitoring of the user in the iDorm. A snapshot of user activity is captured when an actuator, such as those attached to the opening of a window, are altered. The sensor values and the actuator values are recorded at intervals across a defined period. This forms a set of mapped multi-input multi-output data pairs. Using these pairings, the AOFIS system uses a double-clustering approach combining Fuzzy-C Means (FCM) and hierarchical clustering to extract fuzzy membership functions. The membership functions produced are merged with a process to extract rules for defining the users behaviour.

The approach is based upon the enhanced WM method (Wang & Mendel 1992) created by Wang (Wang 2003). Wang's extended WM method is used to construct fuzzy rules through the use of numerical data. Using Wang's method, Doctor et al. gain fuzzy rules relating to the input-output data within the IE. Once the membership functions and the fuzzy rules are captured using the double clustering and rule extraction methods, the agent FLC are entrusted to start controlling the environment on behalf of the user. The agent monitors the users environment and affects actuators based on what has been learned.

Online Adaptation: The AOFIS (Doctor et al. 2005) system also uses an online adaptation and life-long learning system. The user may make adjustments to tune the system, or the behaviour of the user may alter, and in doing so the system adapts to these changes. The rules housed within the system are adapted or new rules are added to take into account these changes to the user preferences. The incorporation of new rules requires the addition of new labelled information directly from the user. This data acts as an enhancement to the original source dataset. The system also incorporates delayed learning in case there are single instances of behaviour. Several occurrences of the same behaviour are needed to trigger a change.

To produce the changes in the rules, the same snapshot method previously employed is again used. If the user overrides the agent system, the snap shot is recorded and passed to the adaptation process. To adapt the rules in the rulebase, the input values are fed into the system to gain a weight. The weight is formed from each rule using the product of the input membership functions as they are fired, for example weight $w > 0$. The rule with the largest consequent membership function are selected to replace the consequent sets of the fired rules, or if no rule is fired at all (Doctor et al. 2005).

Embracing a life long learning strategy in a similar way to Doctor et al. , Acampora et al. (Acampora et al. 2010) present a multi-agent fuzzy strategy to generate context-aware data. The system proposed by Acampora et al. has two levels of adaptation, hardware and software. The hardware level adapts the services offered through the collection of hardware items that are used within the IE. The software induces, from inputs within the IE structure, the most suitable service or collection of services to satisfy the users requirements. Each service provides a response to a users requirements such as temperature, lighting and window control. This is produced through three service concepts: policy, context, and fuzzy context situation.

The policy concept is a rule that determines the level of Quality Of Service (QoS) that is provided, for example a $s_1 = \textit{temperatureControl}$, $P_1 = \{\textit{Low,Medium,High}\}$ where s equals service and P equals policy (Acampora et al. 2010). The concept policy uses context information to help inform the policy concept. This is based on generalised contextual information. The fuzzy context situation allows for context information to be represented at any point in time. The contextual data is combined with temporal information to be processed via predefined membership functions. The membership functions are used to create fuzzy rules. This is achieved through the use of a two pronged learning strategy: *Learning Mode* and *Service Mode*.

Doctor et al. and Rutishauser et al. propose the use of multi-agent systems combined with FL for the use in IEs (Doctor et al. 2005, Rutishauser et al. 2005). The basis for Rutishaiser's (2005) work is a framework that uses an unsupervised online real-time learning process. This is used to form a fuzzy rulebase. The goal of the system is to supply the demands of an intelligent building and the users within, meeting their needs, comfort and preferences (Rutishauser et al. 2005). The system implements a variety of sensors into a building to record varying environmental data. These act as the inputs into the control structure. The control is based on two layers. One layer encapsulates the building as a whole focussing on inputs such as humidity, temperature, radiation, illumination and time. The second layer takes variables for each individual room. This grouping focusses on inputs such as light status and day light. The output relates to a single binary value that forms the basis of each rule, for example bring the blind up. Rutishauser et al. propose the use of two types of rule: static and dynamic. A static fuzzy rule holds the fixed requirements of the system, whereas a dynamic rule relates to the preferences of the user (Rutishauser et al. 2005). The static rulebase is produced pre-specified, and not learnt. The learning process produces dynamic rules online. A conflict process is used to emphasise a rule.

The adaptive learning process used within Rutishauser et al.'s system is based upon the use of adaptive fuzzy reinforcement learning (Bonarini 1997) and maximal structure FL rules (Castro et al. 1999) combined with a Truth Maintenance System (TMS) (Doyle 1979). The proposed algorithm uses all of the information provided by the environment to construct a maximal structure rule-base. The availability of new data from the IE either strengthens or adds rules to the rulebase. The framework decides if the newly acquired information should act as a reward or a punishment in order to produce a reinforcement learning process. This is achieved via a process of a rule subsumption system. However, in order for the rulebase to act effectively rules that contradict each other also go through a process of removal.

2.5.2.2 Fuzzy Logic Systems for Prediction

Embodying an application with the ability to predict is the aim of FL and fuzzy classifiers within AmI. The production of a predictive environment is the focus of the work of Akhlaghinia et al. (Akhlaghinia et al. 2007). In their work, Akhlaghinia et al. use a number of techniques to form a prediction of occupancy within an Assisted Living (AL) environment.

Puteh et al. (Puteh et al. 2012) present a Dynamic Power Usage Scheme (DPUS) to control office workers user space through the profiling of workers activity. The system employs a Wireless Sensor Network (WSN) to record activity data within individual offices. This information is combined with information from a monitoring agent embedded in the users PC. The combined information is analysed by a control server. This application forms the decision making process resulting in a response to the information. Actions such as PC activation, or heating and lighting alteration are used. To accomplish the activation process, the application uses a fuzzy strategy. The raw data from the sensor network is transformed into meaningful information regarding the working situation. A four stage process is employed:

1. The data is pre-processed into a compact, efficient form.

2. The information is fuzzified into meaningful categories.

3. Two user profiles are formed: course and fine.

4. A simple rulebase is extracted to form a control scheme for power management.

A trial system was applied to three offices within a university. A reduction in power usage was shown to occur with the introduction of the prototype DPUS over the currently employed system.

2.5.2.3 Interval and General Type-2 Fuzzy Logic Implementations

There is a significant body of work in the area of type-2 FL and its use within IEs. The uncertainty that embodies the dynamic nature of real-world applications can be captured through the use of type-1 FL and structures such as the FLC (for a more expanded discussion on uncertainty and the use of type-1 FL see section 2.3). The use of type-2 FL, in both its general and interval forms, is proposed to be able to encapsulate the uncertainties within real-world applications beyond crisp type-1 fuzzy sets. Hagras et al. in (Hagras et al. 2007) propose that type-1 fuzzy based systems can only handle slight uncertainties within the short term. As a consequence, these systems will degrade over time. For long term uncertainties to be absorbed, such as those that are experienced within environmental conditions and user activity across seasonal variations, there is a need for the use of type-2 fuzzy sets. Hagras et al. propose that through the use of a third dimension and the inclusion of the Footprint Of Uncertainty (FOU), a system designed around type-2 fuzzy sets can both model and handle the short and long term uncertainties. This discussion is beyond the scope of this thesis. For an insight into this the structure of type-2 fuzzy sets, see the work of Mendel and John (Mendel & John 2002). Hagras (Hagras 2007) proposed that the use of an interval type-2 FLC has the potential to handle high levels of uncertainty and overcome the limitations of a type-1 FLC. This is demonstrated with applications in the area of AmI.

Hagras et al. (Hagras et al. 2007) incorporate the use of type-2 FL into IEs by expanding on previous work in (Doctor et al. 2005). They build on the previous learning system, AOFIS, by producing an incremental version, Incremental Adaptive Online Fuzzy Inference System (IAOFIS). The approach is based on an eight phase operation that again uses a one-pass approach for extracting fuzzy rules and learning. IAOFIS produces type-2 membership functions and fuzzy rules, and utilises an interval type-2 FLC to model the users behaviour. The eight phases are:

Phase 1: Data from the user is captured over a specific time period to form input/output associations.

Phase 2: The system learns from the data captured in **Phase 1:**. The users behaviour is modelled using type-1 fuzzy sets and membership functions. This process is based on the double clustering method outlined in (Doctor et al. 2005).

Phase 3: The produced FLC operates the environment to comply with the users preferences.

Phase 4: Short term uncertainties are adsorbed through the adaptation of the FLC rulebase by adding and altering rules.

Phase 5: As the system starts to degrade, the user is again monitored over a set period of time.

Phase 6: The system again learns from the data that is produced, creating interval type-2 Membership Function (MF) and rules.

Phase 7: The system operates within the environment based on the learnt behaviours and preferences.

Phase 8: The FLC adapts to the short term uncertainties but after an extended period the uncertainties that arise due to the environmental conditions are also absorbed via **Phase 5** (Hagras et al. 2007).

Wagner et al. (Wagner & Hagras 2010) enhance the use of type-2 FL within AmI environments by investigating the application of the more complex general type-2 FL sets within real-world applications. They propose the use of multiple interval type-2 fuzzy sets to generate general type-2 fuzzy sets. Wagner et al. demonstrate the use of interval type-2 fuzzy sets to construct general type-2 fuzzy sets within the confines of the University of Essex iSpace AmI testbed. Drawing on the work of Doctor et al. (Doctor et al. 2005) and Hagras et al. (Hagras et al. 2007), Wagner et al. employ the use FCM to extract type-1 fuzzy sets from collected data to model individual sensor devices and generate FL rules to reproduce the users preferences. As previously discussed, type-2 fuzzy systems have been used to negate the deterioration that occurs as the system experiences changes unavoidable within a real-world setting.

The work of Wagner et al. (Wagner & Hagras 2010) discussed the use of zSlice based general type-2 fuzzy sets from existing interval type-2 sets. This allows for a continuous updating of the fuzzy sets while modelling, what Wagner et al. define as, the agreement of the interval type-2 sets over a period of time. The agreement is an area where the interval type-2 fuzzy sets overlap. The creation of zSlice based general type-2 fuzzy sets from multiple interval type-2 fuzzy sets is based on the concept of the representation of certainty. Areas of membership covered by multiple interval type-2 membership functions are less uncertain than those areas that are covered by fewer functions (Wagner & Hagras 2010). The more MF's that overlap at a specific membership value, the more certainty that is assigned to a crisp input.

2.5.3 Discussion

To demonstrate the effective application of the FuzzyTL framework within uncertain, vague and contextually differing environments, Chapter 4 described the implementation of the framework on data produced within an IE. To highlight previous research in this area and establish the use of FL methods in modelling IEs, Section 2.5.2 gave an overview of a number of implementations of FL techniques that have been applied to the varying domains of IEs.

The application of FL, fuzzy modelling and the use of a FIS in IEs have produced strong results in both simulated and experimental environments. In this section a number of implementations highlighting the combination of IEs and FL to learn, classify and predicate required parameters are illustrated. The application of the FuzzyTL framework is shown to be applicable to the problem domain of IEs through the production of a predictive process. In Chapter 4, an in depth view of the experimental application of the FuzzyTL framework will be given. This chapter will draw on the literature reviewed in this section.

In the following chapter, Chapter 3, the framework that forms the main, novel element will be presented. The literature that has been reviewed throughout this chapter forms the background to understanding this methodology, and the subsequent experimentation that supports it.

Chapter 3

Fuzzy Transfer Learning

3.1 Introduction

A facet of learning is the ability to transfer information from one context to another. Information gained through learning can be generalised, absorbing inconsistencies and anomalies. The hypothesis of this thesis is that what has been learnt can be adapted in order to accomplish the new task, building upon and adapting the previous knowledge. It is in this premise that the Fuzzy Transfer Learning (FuzzyTL) framework is based. In this chapter, the major elements of the FuzzyTL methodology are outlined. Firstly, in Section 3.2 an overview of the framework is given. In the following section, Section 3.3, the definitions used in this section are outlined. In Section 3.4 an in depth discussion of the fuzzy transfer of knowledge will be given with a further discussion of the contextual adaptation in Section 3.5. This chapter highlights the major and novel contributions of this research.

3.2 Overview

The FuzzyTL methodology is contained within a framework structure. The key components can be seen in Figure 3.1.

In this structure there are two distinct processes: firstly, the transferring of the fuzzy concepts and their relationships, and secondly, the adaptation of the fuzzy components using knowledge of the application context. In the first stage the system uses a source of labelled data to instigate a learning process. The learning process uses this source data to construct a Fuzzy Inference System (FIS). The structure of the FIS, as discussed in Chapter 2, consists of fuzzy sets and fuzzy rules. The FIS is used to capture the knowledge from the source, and transfer it to the target task. This process of transferring information is a fundamental aspect of the FuzzyTL methodology, and highlights the use of an Informed Unsupervised (IU) (Cook et al. 2012) Transfer Learning (TL) method.

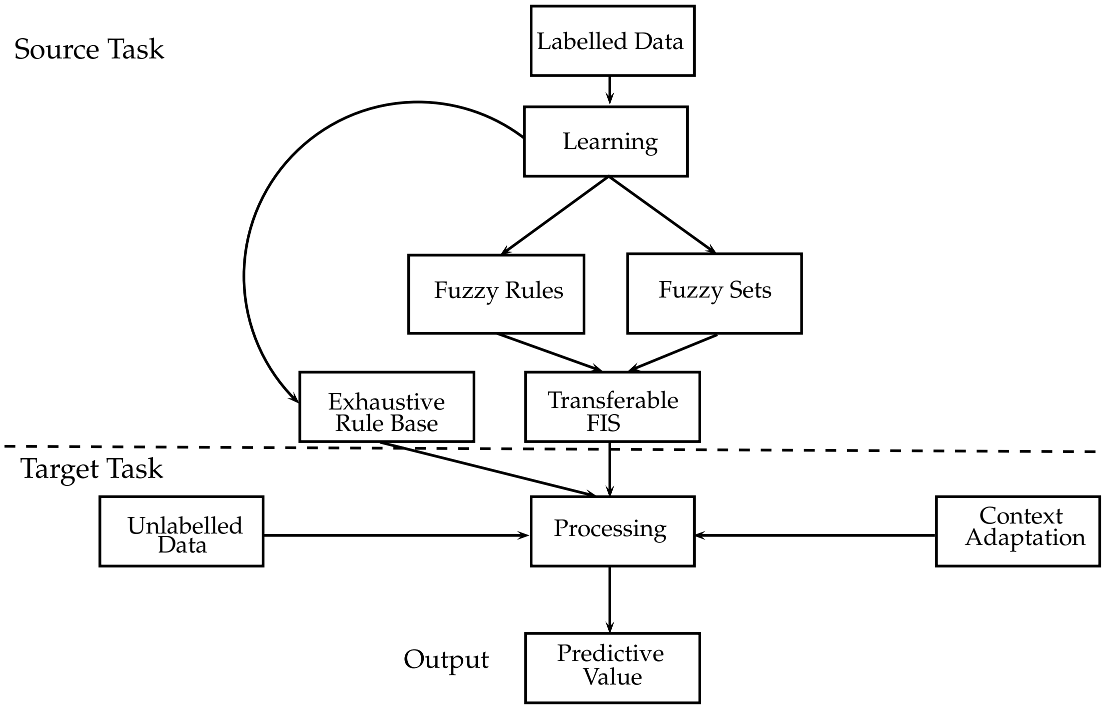

Figure. 3.1. Overview of the Fuzzy Transfer Learning Framework.

Unlike Informed Supervised (IS) TL (as discussed in Section 2.4.3.1) where a quantity of labelled data is required from both the source and the target task, IU TL describes situations where no labelled data is available from the target task. The FuzzyTL methodology captures information from the source task to act as an initial learning point for the target task. This is the basis for the TL process.

The second stage of the framework addresses the adaptation of the FIS. The adaptation process uses knowledge from the unlabelled task dataset coupled with previously learnt information. This process adapts the individual components of the FIS to capture the variations in the data. Alterations and variations from situation to situation, are absorbed through changes made within the domains of the fuzzy sets and adaptations to the rulebase. This is in keeping with the ideas discussed in Section 2.4.3.3 relating to domain adaptation. In this chapter, the FuzzyTL methodology is shown to be able to use the transfer of information to assist in bridging the knowledge gap. Through an online adaptation process, newly accrued information can be absorbed. In the following chapter, Chapter 4, the application of these methods are shown through the frameworks abilities to predict tasks using real-world data sources. The subsequent sections give an in depth explanation of the FuzzyTL framework. In order to understand the following methodology, a number of definitions are set out.

3.3 Definitions

To continue the discussion of the novel FuzzyTL methodology, a series of elements need to be predefined. To reiterate definitions discussed in Chapter 2, a source domain $\mathcal{D}_s$ can be defined as

$$\mathcal{D}_s = \{(x_1^s, x_2^s, y^s)\}_s^N \tag{3.1}$$

where $x \in X$ are data inputs, $y \in Y$ is an output, and N is the number of data tuples within the domain. Equally, the target $\mathcal{D}_t$ and adaptive domains $\mathcal{D}_a$ can be defined in the same way.

Additionally, the domain can be defined through the use of intervals. Within this chapter, an interval is referred to as a bounded set of real numbers

$$A = [a_L, a_R] = \{a : a_L \leq a \leq a_R, a \in \mathbb{R}\} \tag{3.2}$$

where a_L and a_R are the left and right limits of the interval A (Sengupta & Pal 2000). Two intervals A and B are considered equal if their corresponding endpoints are equal. So, $A = B$ if $a_L = b_L$ and $a_R = b_R$. The intersection of two intervals is empty $A \cap B = \emptyset$, if $a_R < b_L$ or $b_L > a_R$ (Moore 1987). The extended addition $\oplus$ and extended subtraction $\ominus$ can be defined as:

$$A \oplus B = [a_L + b_L, a_R + b_R], \tag{3.3}$$

$$A \ominus B = [a_L - b_L, a_R - b_R]. \tag{3.4}$$

The notion of $\leq$ is extended to intervals as

$$A \leq B \iff a_R \leq b_L. \tag{3.5}$$

Set inclusion is also extended to be defined as

$$A \subseteq B \iff a_L \geq b_L \text{ and } a_R \leq b_R. \tag{3.6}$$

Based on this definition and using the interval notation, a source domain interval can be defined as $\mathcal{D}_s^{\mathcal{I}} = \{[x_{1L}^s, x_{1R}^s], [x_{2L}^s, x_{2R}^s], [{y^s}_L, {y^s}_R]\}$.

A domain will also be defined through its relationship to fuzzy sets. A source domain with two inputs, and a single output is defined as

$$\mathcal{D}_s = \{{}^fX_1^s, {}^fX_2^s, {}^fY^s\} \tag{3.7}$$

where ${}^fX_1^s$ and ${}^fX_2^s$ are sets of fuzzy input sets, and ${}^fY^s$ is a set of fuzzy output sets. ${}^fX_1^s$ can be defined as

$$^{f}X_1^s = \{vs^{f x_1^s}, s^{f x_1^s}, m^{f x_1^s}, l^{f x_1^s}, vl^{f x_1^s}\} \tag{3.8}$$

where vs to vl are the sets *very small*, *small*, *medium*, *large* and *very large* respectively. These sets can be any description that is suitable to the context. Within this chapter, the fuzzy sets are constructed as normal, continuous and triangular.

The rulebases used within the subsequent sections are defined using the same notation. A rulebase that contains two antecedent and one consequent sets is depicted as

$$\mathcal{R} = \{^{f}X_1^r, ^{f}X_2^r, ^{f}Y^r\}^P \tag{3.9}$$

where R is the rulebase, X is a data input, Y is the corresponding output, and P is the number of rules.

3.4 Transferring Fuzzy Concepts

The first stage of the FuzzyTL process is the construction of the FIS (Fuzzy Inference System). Fuzzy rules and fuzzy sets are formed via the use of an Ad-Hoc Data Driven Learning (ADDL) process which is calculated from numerical data. The method uses numerical data to form the sets and rules, a procedure based on an algorithm proposed by Wang-Mendel (WM) (Wang & Mendel 1992). The FuzzyTL framework builds on the method by adding a novel rule reduction process. The addition of a fuzzy frequency measure reduces the impact of anomalous data, and increases the information extracted from the numerical data.

3.4.1 Transferring Knowledge Through a Fuzzy Logic System

The basis of the WM process is the formation of fuzzy sets and fuzzy rules from numerical data as outlined in Section 2.3.5.1. The use of the method is not restricted to an individual application domain and has been shown to be applicable to a broad number of implementations (Teodorovic et al. 2001, Yang et al. 2010, Doctor et al. 2005). The methodology is a generalised one and is in keeping with the basis of the FuzzyTL framework. The algorithm produced by WM can be described as an ad hoc data-driven method. Benefits can be attributed to using this method of extraction. Its simplicity makes it easily understandable and the nature of the low computation required allows for a greater speed of implementation. The swift preliminary fuzzy modelling process allows for the subsequent adaptation of the model by other methods (Casillas et al. 2000). The proposed fuzzy frequency pruning builds on these attributes, allowing for a greater depth of information to be extracted from the numerical data.

In order to transfer knowledge from one context to another, this information initially needs to be captured in a model. Within the FuzzyTL framework, this model is a FIS. As discussed in Section 2.3.4, the FIS consists of fuzzy sets and fuzzy rules. To form both the sets and rules of the FIS, the WM methodology uses numerical data to extract the structure. The source task is provided as a numerical dataset consisting of labelled data. Non-numerical data can be used within the system but will require pre-processing. For example, the use of categorical data types such as hair colour (brown, red, blonde and black) or car makes (Ford, Audi and Citreon) can be adjusted into real-valued numerical values.

The labelled source dataset is required to be input-output pairs. Previously defined, these can be Multiple Input Multiple Output (MIMO) data instances. The FuzzyTL framework follows the standard WM process for the production of the fuzzy sets and fuzzy rules. In the WM method, a standard $2n + 1$ quantity of sets are used. This is a user defined number and dependent on expert knowledge. It can be adapted for changing implementations and situations. In this implementation, triangular functions will be used, however the use of singleton, trapezoid or Gaussian functions can be incorporated into the system.

Following the production of the fuzzy sets, the WM method (see Section 2.3.5.1), requires the production of a fuzzy rulebase. A fundamental aspect of this process is the reduction in quantity of the rules in the exhaustive rulebase. Under the standard WM process, a weighted measure is used to produce a Reduced Rule Base (RRB). This emphasises the rules with the highest antecedent and consequent membership values. In the exhaustive rulebase, there exists further information that can inform both the RRB construction and further enhance the transfer process. The FuzzyTL framework adds to the WM method by supplementing the process with a fuzzy frequency measure. The addition of the fuzzy frequency measure endeavours to remove the possibility of anomalous data influencing the production of fuzzy rules.

The following section outlines the extension of the numerical extraction process to incorporate the frequency rule pruning method.

3.4.2 Extending the Wang-Mendel Method: Fuzzy Frequency Rule Pruning

To highlight the properties of the FuzzyTL framework, an example will be used. The example can be defined as Multiple Input Single Output (MISO) data consisting of a two input (x_1,x_2) and one output (y) example, with m data points.

In the standard WM process, the rulebase is created by using the membership values of each data instance. The inputs x_1, x_2 and output y each produce a value based upon the greatest membership in each set of the domain. Using these membership values, the corresponding sets form the basis of the rule. Each data instance, as a result, produces a single rule. This is discussed at greater length in Section 2.3.5.1. To reduce the rulebase, rule pruning is carried out. Each rule is assigned a weight based on the membership of the antecedent and consequent values. Figure 3.2 shows two instances of data and the corresponding membership values.

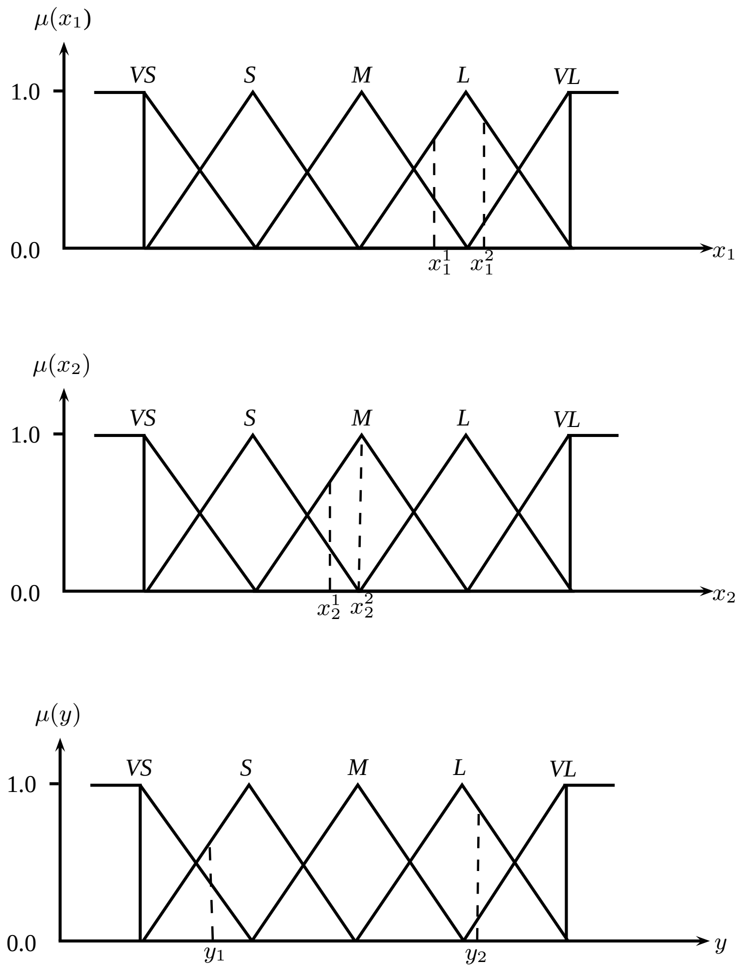

Figure. 3.2. Membership Values in Fuzzy Sets x_1, x_2 and y

The resulting rules each produce differing consequent values. These can be described as:

$(x_1^1, x_2^1, y^1) \Rightarrow [x_1^1$ (0.65 in L), x_2^1 (0.7 in M), y_1 (0.55 in S)$] \Rightarrow$ Rule 1

IF x_1^1 is L and x_2^1 is M THEN y_1 is S;

$(x_1^2, x_2^2, y^2) \Rightarrow [x_1^2$ (0.75 in L), x_2^2 (1.0 in M), y_2 (0.75 in L)$] \Rightarrow$ Rule 2

IF x_1^2 is L and x_2^2 is M THEN y_2 is L;

Based on this premise, m tuples produce the equivalent quantity of rules in the rulebase. As a result, large datasets can produce large, unmanageable and highly inefficient rulebases. Multiple cases of the same rule can occur with multiple rules sharing the same antecedent values. To remove this impact, a rule pruning process is used.

The WM process uses a rule reduction method based on a weighting of the output. The weight is formed from the membership of each antecedent and consequent function. The basis of this approach is to maximise the output of the rule in terms of the membership. Figure 3.2 shows the product of $\mu(x_1^2, x_2^2, y^2)$ is greater than that of $\mu(x_1^1, x_2^1, y^1)$. To remove the conflict produced resulting from sharing the antecedent values, the lower valued rule is removed. By removing each rulebased on the greatest weight, the rule with the maximum membership based on the antecedent group remains.

To capture more of the information contained within the exhaustive rulebase, the FuzzyTL framework uses a frequency value. This value gains more information from the original rulebase by focusing on the number of occurrences of each rule.

The initial step of the process is to capture the minimum and maximum of the frequency

$$\begin{aligned} F_{min} &= \sup_{r \in \mathcal{E}} F_{Freq}(r) \\ F_{max} &= \inf_{r \in \mathcal{E}} F_{Freq}(r) \end{aligned} \tag{3.10}$$

where r is a rule in the exhaustive rulebase $\mathcal{E}$, and F_{min} and F_{max} are the quantity of the lowest and highest occurring rules.

Based on this range of values, a membership function is formed to capture the frequency of the rules $\mu_{Freq}(F)$ within the rulebase. This can be described as

$$\mu_{Freq}(F) = e^{\frac{-(F_r - c)^2}{2\sigma^2}}. \tag{3.11}$$

The parameter c uses the value F_{max} previously taken from the frequency of the rule. This is the peak of the function and provides the point of the maximum membership. σ is a predefined value used to shape the function. The input to the function is taken as the frequency of each rule F_r. As the values to be used within the function can be at the extremes of the interval, the use of a triangular function can result in a zero value. To negate this, a Gaussian function is implemented.

The membership value, $\mu_{Freq}(F)$, provides an additional weighting to the process used by WM to prune the rulebase. The memberships of the antecedent and consequent values are combined with the fuzzy frequency to form a new weight.

As the rules are grouped into sections based on the similarity of the antecedent values of the rules, the rule with the highest overall weighting is retained in the RRB. Algorithm 3.1 illustrates

Algorithm 3.1 Process to Gain the Highest Weighted Fuzzy Frequency Rule.

Exhaustive Rule Base $\mathcal{E}$
Input Variable $x \in X$
Number of Input Variables g
Output Variable $y \in Y$
Number of Output Variables h
Rule r
Retained Rule RR
Reduced Rule Base $\mathcal{R}$
Frequency of a Rule F_{r_i}
Rules Grouped by Antecedent GR
Number of Rule in Group N_{GR}
Current Weighting $\mathcal{CW}$
Weighting $\mathcal{W}$
$\mathcal{CW} = 0$
for $i = 1; i < N_{GR}; i{+}{+}$
  $\mathcal{W} = \prod_{m=1}^{g} \mu_X(x_{r_i}^m) \prod_{n=1}^{h} \mu_Y(y_{r_i}^n)\mu_{Freq}(F_{r_i})$
  if $\mathcal{W} > \mathcal{CW}$ **then**
    $RR = r_i$
    $\mathcal{W} = \mathcal{CW}$
  end if
end for
$\mathcal{R} \stackrel{+}{=} RR$

the process. Here, the largest weight is calculated for a single group of rules with equal antecedent values. The same process would be carried out for all groups throughout the dataset.

Initially, the algorithm combines the membership of the antecedent and consequent values for each rule that exists within the group GR. The fuzzy frequency value is used to add a weight to the quantity. The highest value is found by iterating through the grouped rules. The final defined rule is added to the Reduced Rule Base $\mathcal{R}$.

To illustrate the process a toy example will be presented. Table 3.1 highlights 10 data points used in this example.

Each data tuple produces a single rule. This results in the grouped rulebase consisting of 10 rules. The standard WM method groups together the antecedent values in order to eliminate duplication. As a result, the example outlined would be brought together as a single group. To remove matching rules a weight is formed. In the example, the rule with the greatest product membership outputs a rule of $\{$*Very Low, Very Low, Very Low*$\}$. This is shown in Table 3.1 as rule 1. For the rule described, the value is formed as

Rule	$A(x_1)$	$\mu(x_1)$	$A(x_2)$	$\mu(x_2)$	$A(y)$	$\mu(y)$	Freq. of Rule	$\mu(Freq)$	WM	With Freq.
1	Very Low	0.30	Very Low	0.60	Very Low	0.75	1	0.13	0.13	0.02
2	Very Low	0.30	Very Low	0.60	Low	0.25	6	1.00	0.04	0.04
2	Very Low	0.30	Very Low	0.60	Low	0.25	6	1.00	0.04	0.04
2	Very Low	0.30	Very Low	0.60	Low	0.25	6	1.00	0.04	0.04
2	Very Low	0.30	Very Low	0.60	Low	0.25	6	1.00	0.04	0.04
2	Very Low	0.30	Very Low	0.60	Low	0.25	6	1.00	0.04	0.04
2	Very Low	0.30	Very Low	0.60	Low	0.25	6	1.00	0.04	0.04
3	Very Low	0.30	Very Low	0.60	Med	0.00	2	0.28	0.00	0.00
3	Very Low	0.30	Very Low	0.60	Med	0.00	2	0.28	0.00	0.00

Table 3.1. Toy Example Data for Fuzzy Frequency Rule Pruning.

$$\begin{aligned}
\mathcal{W} &= \mu_{X_1}(x_1^1) \times \mu_{X_2}(x_2^1) \times \mu_Y(y^1) \\
\mathcal{W} &= 0.30 \times 0.60 \times 0.75 \\
\mathcal{W} &= 0.13
\end{aligned} \tag{3.12}$$

However, this rule is based on the lowest frequency of the occurrence, implying a low number of instances of this rule type occurring in the dataset. The frequency membership emphasises the number of data tuples that formed the rule. To enhance the WM pruning, the frequency of each rule is added to this value. This value is calculated using a Gaussian function (see Equation 3.11). Using toy example presented here, the c is defined as $F_{max} = 6$ and

$$\begin{aligned}
\sigma &= (F_{max} - F_{min}) \times dT \\
\sigma &= (6 - 1) \times 0.5
\end{aligned} \tag{3.13}$$

where dT is a defined percentage that defines the shape of the Gaussian function.

Combining the antecedent, consequent and fuzzy frequency values together for the same rule that previously produced the highest value, a different rule is given.

$$\begin{aligned}
\mathcal{W} &= \mu_{X_1}(x_1^1) \times \mu_{X_2}(x_2^1) \times \mu_Y(y^1) \times \mu_{Freq}(F^1) \\
\mathcal{W} &= 0.30 \times 0.60 \times 0.75 \times 0.13 \\
\mathcal{W} &= 0.02.
\end{aligned} \tag{3.14}$$

Using the frequency measure, the lower valued WM rule $\{$*Very Low, Very Low, Low*$\}$ is kept.

$$\begin{aligned} \mathcal{W} &= \mu_{X_1}(x_1^2) \times \mu_{X_2}(x_2^2) \times \mu_Y(y^2) \times \mu_{Freq}(F^2) \\ \mathcal{W} &= 0.30 \times 0.60 \times 0.25 \times 1.00 \\ \mathcal{W} &= 0.04. \end{aligned} \tag{3.15}$$

The use of the fuzzy frequency measure produces a greater value, (0.04) than the highest value using the WM method alone (0.02). By combining both a strength and frequency measure, a single anomalous rule with a high strength will have a reduced influence. Rules that are frequently produced but have a very low strength equally are unable to heavily influence the process, and as a result will not be placed into the rulebase.

3.5 Adaptation Through Learning

As outlined in Section 3.4, the transferral of the FIS embodies the TL component of the FuzzyTL methodology. Using the notation in Section 2.4.1, this can be described as transferring a source domain $\mathcal{D}_s$ to model a predictive function of a target domain $\mathcal{D}_t$. The production of the FIS can be referred to as a learning task $\mathcal{T}_s$. The source domain $\mathcal{D}_s$ can be depicted as containing a triple of values $(x_1 \in X_1, x_2 \in X_2, y \in Y)$. A domain with n instances of data can be represented as $\mathcal{D}_s = \{(x_1, x_2, y)\}_s^N$. The target domain consists of unlabelled data, and can be represented as $\mathcal{D}_t = \{(x_1, x_2)\}_t^N$. The relationship between $\mathcal{D}_s$ and $\mathcal{D}_t$ influences the output of the model. If there exists some relationship, explicit or implicit, between the two domains this is categorised as being *related*. This is further discussed in Section 2.4.1. The nature of the relationship will dictate the necessity for the adaptation of the knowledge contained within the source domain and the learning task. If the domains are equal and the learning tasks are approaching the same problem, no adaptation is required, however, this is rare within real world applications. Separation of the domains results in the need for an adaptation process. This form of transfer is defined as transductive. According to Cook et al (Cook et al. 2012), the formation of the transfer process can also be described as IU learning (see Section 2.4 for further details).

In order for the framework to absorb such changes from the source to the target contexts, the elements of the transferable FIS are adapted. Using the knowledge housed within the exhaustive rulebase, the FIS itself and newly acquired information, changes are made in order for the framework to output the required data. The adaptation consists of five interlaced stages.

1. **External Input Domain Adjustment:** The adaptation of the periphery of the input domain through information from the *target* task.

2. **Internal Input Domain Adjustment:** The adaptation of the internal aspects of the input

domain through information from the *source* and *target* task.

3. **Output Domain Adaptation:** The adaptation of the output domain based upon a mapping between the *source* task and the *target* task.

4. **Rule Base Modification via Source Rule Comparison:** Modification of the rulebase using comparison measures between the *source* task and the *target* task rule structures.

5. **Rule Adaptation Using Euclidean Distance Measure:** The creation of new additional rules composed of knowledge gathered from the *source* task and newly acquired data.

Each of the five stages approaches a separate element of the FIS or an issue that arises in the transferring of a FIS. The initial three stages focus on the adaptation of the domains that make up the sets. As information is gained from each of the data instances, both the antecedent and consequent sets are altered to better fit the newly acquired data. The fourth and fifth stages concentrate on the adaptation of the rulebase. The source data is used as the foundation for the changes required within the rulebase. Taking each stage in turn, the following sections will describe in detail the nature of the processes.

3.5.1 External Input Domain Adjustment: Stage One

As discussed in Section 2.4.3.3, a knowledge gap can occur during the transfer of learning structures from one contextual domain to another. This can be captured as both differences in the domains themselves, and differences in the learning structure.

Focussing on these concepts, a simple analogy can be introduced to help explain the adaptation process. An individual is taught how to ride a bicycle. They use the bicycle to ride the short journey to work each day. Their place of work moves, increasing the distance they need to travel. The same skills are applicable across the two tasks, however the domain has altered. Both are road cycling, however one task is significantly further. A knowledge gap is produced as a result. In such a case, adaptation of the domain is required in order to incorporate the skills.

To absorb such contextual differences in the source and target tasks, the FuzzyTL adopts a process of adapting the minimum and maximum values within the domain. Taking each input instance of the dataset, the framework adjusts the range of the interval according to any difference calculated between the transferable FIS and the new input values. The result is the adaptation of the sets that form the basis of the FIS.

Based upon the example presented in the previous section, the $\mathcal{D}_s$ would consist of data tuples (x_1, x_2, y), where x_1 and x_2 are inputs and y represents the output. A new domain is formed based on $\mathcal{D}_s$ incorporating the alterations made through the adaptation process. This is defined as $\mathcal{D}_a$. $\mathcal{D}_a$ represents missing information that can occur between source and target tasks. Each input and

Algorithm 3.2 Adaptation Algorithm: Stage One External Input Domain Adjustment

Adaptation Process Step 1: External Input Domain Adjustment
Input Variable $x \in X$
Output Variable $y \in Y$
Number of Data Tuples N
Source Domain $\mathcal{D}_s = \{(x_1, x_2, y)\}_s^N$
Source Interval Domain $\mathcal{D}_s^{\mathcal{I}} = \{[x_{1L}, x_{1R}], [x_{2L}, x_{2R}], [y_L, y_R]\}$
Adaptation Interval Domain $\mathcal{D}_a^{\mathcal{I}} = \{[x_{1L}, x_{1R}], [x_{2L}, x_{2R}], [y_L, y_R]\}$
Target Domain $\mathcal{D}_t = \{(x_1, x_2, y)\}_t^N$

$\mathcal{D}_a^{\mathcal{I}} = \mathcal{D}_s^{\mathcal{I}}$
$i = 1$
while $i < N$ **do**
  if $\mathcal{D}_t(x_1)^i < \mathcal{D}_s^{\mathcal{I}}(x_{1L})$ **then**
    $\mathcal{D}_a^{\mathcal{I}}(x_{1L}) = \mathcal{D}_t(x_1)^i$
  else if $\mathcal{D}_t(x_1)^i > \mathcal{D}_s^{\mathcal{I}}(x_{1R})$ **then**
    $\mathcal{D}_a(x_{1R}) = \mathcal{D}_t(x_1)^i$
  end if

  if $\mathcal{D}_t(x_2)^i < \mathcal{D}_s^{\mathcal{I}}(x_{2L})$ **then**
    $\mathcal{D}_a^{\mathcal{I}}(x_{2L}) = \mathcal{D}_t(x_2)^i$
  else if $\mathcal{D}_t(x_2)^i > \mathcal{D}_s(x_{2R})$ **then**
    $\mathcal{D}_a^{\mathcal{I}}(x_{2R}) = \mathcal{D}_t(x_2)^i$
  end if
end while

output variable in the domain is represented as an interval. For example, $\mathcal{D}_s$ can be described as

$$\begin{aligned} \mathcal{D}_s(X_1) &= [x_{1L}, x_{1R}] \\ \mathcal{D}_s(X_2) &= [x_{2L}, x_{2R}] \\ \mathcal{D}_s(Y) &= [y_L, y_R]. \end{aligned} \tag{3.16}$$

To capture the knowledge within the data from the target domain $\mathcal{D}_t$, each input data point is analysed. The input interval is adapted if the value extends beyond the left (x_L) or right (x_R boundaries. This produces a new set structure. Algorithm 3.2 shows this process for two inputs.

In Algorithm 3.2, the left and right limits of the interval are adapted based on data from the target domain. As new unlabelled data is received from the target, this stage compares the input values to those in the source. If the value is less than the source left limit $\mathcal{D}_s^{\mathcal{I}}(X_L)$, then the adaptive domain left limit ${\mathcal{D}_a}^{\mathcal{I}}(X_L)$ is decreased to the same value. If the value is greater, the right interval limit $\mathcal{D}_a^{\mathcal{I}}(X_R)$ is increased to the target value $\mathcal{D}_t(X)^i$. By applying this process to the domain, the sets that are contained are altered.

In Figure 3.3 the sets are equally spaced. Any adaptation to the domain results in an equal distribution across the sets. This is due to the equal spacing. Extension of the domain requires a simple change to the footprint of each set. Figure 3.3 shows an expansion across the domain using the new X_L and X_R values.

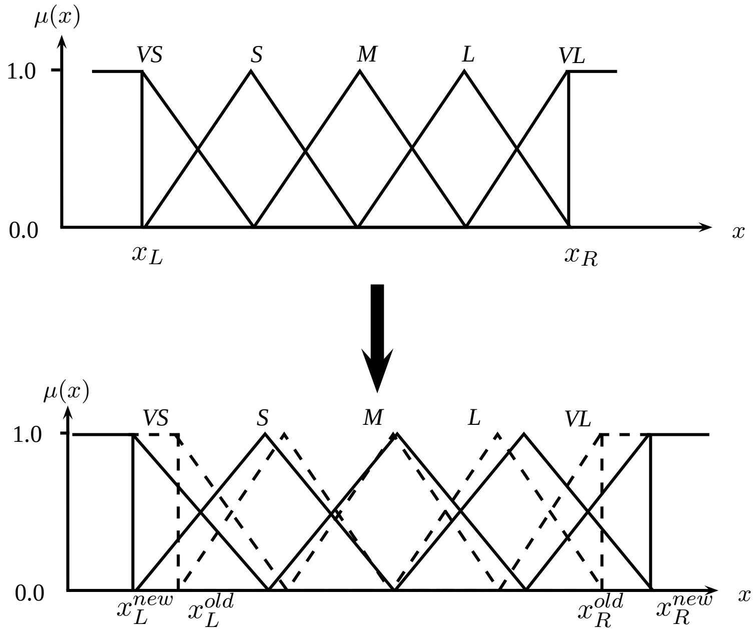

Figure. 3.3. External Adaptation of Sets Based on New x_L and x_R Values.

Unevenly distributed membership functions require a scaling function in order to adapt the sets. In the FuzzyTL framework triangular functions are used, however, other functions are applicable. A triangular function can be defined as:

$$A(x) = \begin{cases} \left(1 - \frac{|x-a|}{s}\right) & \text{when } a - s \leq x \leq a + s \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

where x is the input value, a is the centre of the function, b is the height and s is the width. In a similar manner to equally distributed sets, the centre of the function a is used as the anchor point. If the domain is shifted in a negative or positive direction, the sets are moved by the centre points. Each point is moved an equal distance. Any extension or compression of the domain requires that the sets are altered according to the scaling. This process is shown in Figure 3.4. Here, three sets are shown in the X universe. The sets shown (*Small*, *Medium* and *Large*) are uneven and have differing footprints. The example shows the domain increased by 30%. This results in a similar alteration to the sets. The percentage domain increase alters the width and centre of the sets. Taking the *Large* set, the width of the set is 2. Following the increase in the domain, this is increased to 3. Equally the centre is shifted from 4 to 6.

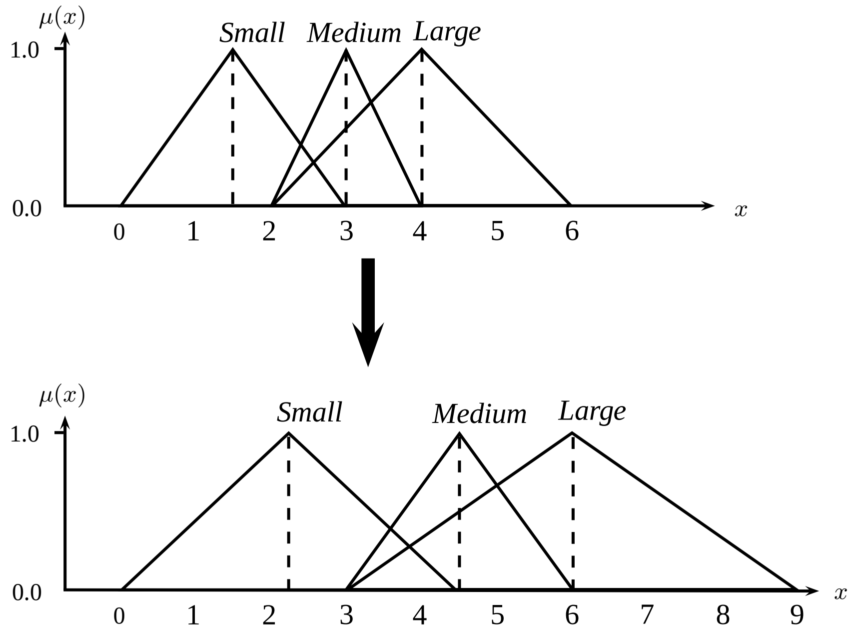

Figure. 3.4. Adaptation of Unevenly Distributed Sets Based on New Minimum and Maximum Input Values.

If the domain of the target task is contained within the source task, an alternative strategy is required to adapt both set structure and domains. Stage Two approaches this issue through the use of an online adaptation approach, coupled with knowledge from the transferral of the fuzzy logic elements.

3.5.2 Internal Input Domain Adjustment: Stage Two

The second adaptation stage also focuses on the input domains. The transferring of source domains can require adaptation to remove the knowledge gap. The knowledge gap can be represented by differences in the domain intervals. In Figure 3.5, interval $\mathcal{D}^{\mathcal{I}}_{t_1}$ is shown to be a subset of $\mathcal{D}^{\mathcal{I}}_{s}$, $\mathcal{D}^{\mathcal{I}}_{t_1} \subset \mathcal{D}^{\mathcal{I}}_{s}$. $\mathcal{D}^{\mathcal{I}}_{t_2}$ partially overlaps $\mathcal{D}^{\mathcal{I}}_{s}$. This can be represented as $\mathcal{D}^{\mathcal{I}}_{s_L} < \mathcal{D}^{\mathcal{I}}_{t_{2L}} < \mathcal{D}^{\mathcal{I}}_{s_R}$ and $\mathcal{D}^{\mathcal{I}}_{s_R} < \mathcal{D}^{\mathcal{I}}_{t_{2R}}$. Where necessary, stage one increases the overall size of the domain interval either by decreasing the left limit or increasing the right limit to reduce the differences. However, in transferring the source to the target, there may be a need to reduce the domain to within the source extremities, either partially or wholly. In Figure 3.5, the source domain $\mathcal{D}^{\mathcal{I}}_{s}$, has been reduced to form $\mathcal{D}^{\mathcal{I}}_{t_1}$. $\mathcal{D}^{\mathcal{I}}_{t_2}$ is shifted in a positive direction along the axis. The left limit $\mathcal{D}^{\mathcal{I}}_{t2_L}$ has been moved in a positive direction from $\mathcal{D}^{\mathcal{I}}_{s_L}$. This is accomplished by stage two of the adaptation process. The right limit $\mathcal{D}^{\mathcal{I}}_{t2_R}$ has also been shifted in a positive direction. This outside of the source interval.

This is accomplished by stage one.

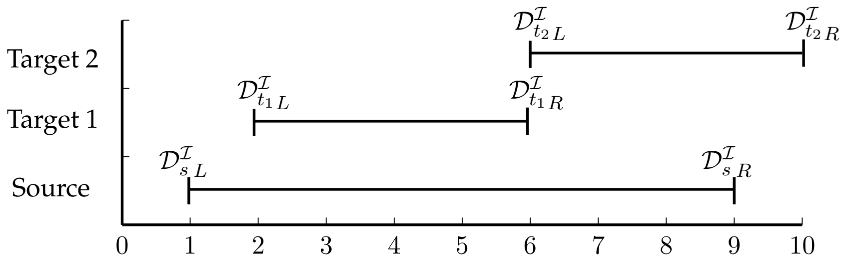

Figure. 3.5. Example of Internal Domain Containment.

To produce internal domain change, the second adaptation stage uses information transferred from the source domain. This can be illustrated by a toy example. Table 3.5.2 represents 20 data instances taken from a single input value x. The three stage procedure is shown below.

	Source		Target Value	Target Local Values		Fuzzy Membership			Adaptation	
Data	$\mathcal{D}^{\mathcal{I}}_s(x_L)$	$\mathcal{D}^{\mathcal{I}}_s(x_R)$	$\mathcal{D}_t(x)$	$\mathcal{D}^{\mathcal{I}}_t(x_L)$	$\mathcal{D}^{\mathcal{I}}_t(x_R)$	Min Proximity	Max Proximity	System Influence	$\mathcal{D}^{\mathcal{I}}_a(x_R)$	$\mathcal{D}^{\mathcal{I}}_a(x_R)$
1	90.00	500.00	140.00	140.00	140.00	0.93	0.03	N/A	90.00	500.00
2	90.00	500.00	180.00	140.00	180.00	0.93	0.06	N/A	90.00	500.00
3	90.00	500.00	215.00	140.00	215.00	0.93	0.11	N/A	90.00	500.00
4	90.00	500.00	310.00	140.00	310.00	0.93	0.38	N/A	90.00	500.00
5	90.00	500.00	345.00	140.00	345.00	0.93	0.52	N/A	90.00	500.00
6	90.00	500.00	370.00	140.00	370.00	0.93	0.63	N/A	90.00	500.00
7	90.00	500.00	395.00	140.00	395.00	0.93	0.74	N/A	90.00	500.00
8	90.00	500.00	415.00	140.00	415.00	0.93	0.82	N/A	90.00	500.00
9	90.00	500.00	455.00	140.00	455.00	0.93	0.95	Lesser	90.00	455.00
10	90.00	500.00	460.00	140.00	460.00	0.93	0.96	Greater	90.00	460.00
11	90.00	500.00	450.00	140.00	460.00	0.93	0.96	N/A	90.00	460.00
12	90.00	500.00	410.00	140.00	460.00	0.93	0.96	N/A	90.00	460.00
13	90.00	500.00	380.00	140.00	460.00	0.93	0.96	N/A	90.00	460.00
14	90.00	500.00	275.00	140.00	460.00	0.93	0.96	N/A	90.00	460.00
15	90.00	500.00	250.00	140.00	460.00	0.93	0.96	N/A	90.00	460.00
16	90.00	500.00	220.00	140.00	460.00	0.93	0.96	N/A	90.00	460.00
17	90.00	500.00	185.00	140.00	460.00	0.93	0.96	N/A	90.00	460.00
18	90.00	500.00	155.00	140.00	460.00	0.93	0.96	N/A	90.00	460.00
19	90.00	500.00	125.00	125.00	460.00	0.97	0.96	Greater	125.00	460.00
20	90.00	500.00	100.00	100.00	460.00	0.99	0.96	Greater	100.00	460.00

Table 3.2. Example Data for Inner Domain Adaptation.

Step One: Initialisation

- The process relies on the use of information gathered from the source task. This data can be seen in the columns two and three of Table 3.5.2. To gain the source input interval, the whole dataset is processed. The input interval can be defined as $\mathcal{D}^{\mathcal{I}}_s(X) = [x_L, x_R] = [90.00, 500.00]$. By calculating this value, it allows for the target input values to be compared to the source.

Step Two: Correlation

- Unlike the source task, the target task has extremely limited data availability. To address this lack of knowledge, the adaptation system uses local minimum and maximum values to compare to the source values. As the target task acquires data points, local minima and maxima are calculated. Using Table 3.5.2 as an example, at **Data 10** a local maxima is calculated and at **Data 19** a local minima. If one, or both of these values fall within the interval that is represented by the source values x_L and x_R, a proximity measure is produced to ascertained whether the domain is adapted. The proximity is based upon a membership function. The function can take any form chosen, although within the FuzzyTL framework a Gaussian function is used (see Equation 3.11). The membership function is based on the source input domain interval. Based on the example, this is defined as

$$\begin{aligned} &\mathcal{D}_s^{\mathcal{I}}(X) = [x_L, x_R] \\ &x_L = 90.00 \\ &x_R = 500.00 \end{aligned} \tag{3.18}$$

  Using a predefined threshold to act as a benchmark, the system calculates the proximity of the target value to the source value. The corresponding mapped input value is updated within the target. Using the data in Table 3.5.2 a simple example can be illustrated. At **Data 1**, the target value $\mathcal{D}_t(x)$ produces a target local minimum represented as $\mathcal{D}_t^{\mathcal{I}}(x_L)$ and maximum $\mathcal{D}_t^{\mathcal{I}}(x_R)$ that are within the source interval $\mathcal{D}_s^{\mathcal{I}}(X)$. As $\mathcal{D}_t(x)$ increases, the local maximum corresponds. At **Data 7**, $\mathcal{D}_t^{\mathcal{I}}(x_R) = 395.00$. Using the proximity function constructed using the Gaussian definition, a maximum membership is given when the value is closest to the defined centre. Focussing on the example, the local maximum at **Data 7** returns a membership value of 0.74 to the maximum proximity of the source value. In this example, a predefined threshold of 0.95 is placed on the membership values. This value is user defined. When the threshold is reached, adaptation of the domain can occur. **Data 9** shows the target local maximum as represented by $\mathcal{D}_t^{\mathcal{I}}(x_R)$ returning a maximum proximity membership of 0.95, which is equal to the threshold. This would result in the adaptation taking place.

Step Three: Negative Influence

- Adaptation of the input domains is monitored based on its impact. **Data 9** shows the threshold of the maximum proximity being reached. This results in the input domain $\mathcal{D}_a^{\mathcal{I}}(x_R)$ being adapted from 500.00 to 455.00. To ascertain the influence of this adaptation, a comparison is made between the maximum membership of the rulebase previous to the update, to the same value following the changes. A reduction in value returns the system to its previous state. This allows for the system to police the adaptation, and endeavour to move away from a state of negative transfer.

The inner adaptation of the fuzzy sets can be seen in Figure 3.6. In this example, the sets of X are compressed based on the adapted x_L and x_R limits. These values are within the original source domain interval.

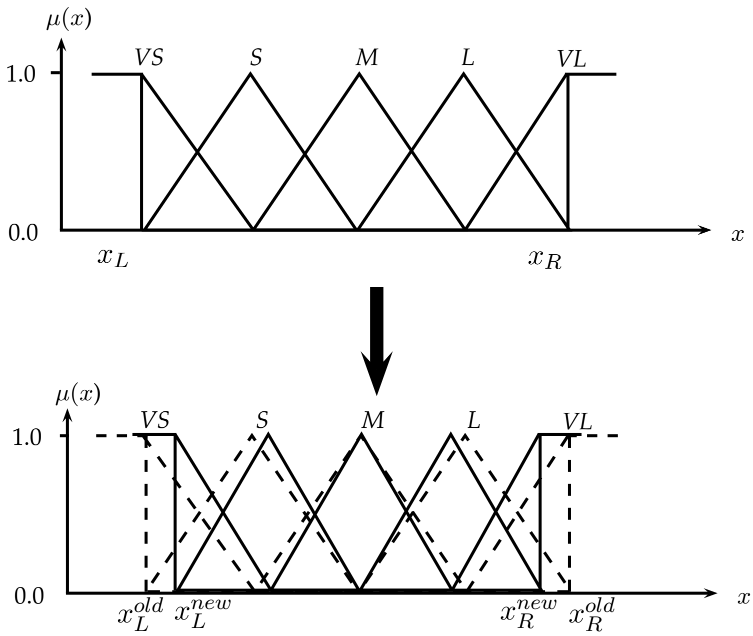

Figure. 3.6. Internal Adaptation of Sets Based on New x_L and x_R Values.

The first two stages of the adaptation focus on the input variable domains and as a result the antecedent sets. Data is available to produce adaptation within these domains. The unlabelled nature of the data impedes the ability for direct adaptation of the target consequent domains. The third adaptation stage combines data produced by the framework with new task information to approach this problem. The domains are adapted in an online process. Online learning is discussed further in Section 2.3.9. The five adaptation stages use data that is produced within the target task to update the model produced and transferred from the source data. The use of data from the target task to alter the model is the foundation of adaptive online learning.

3.5.3 Output Domain Adaptation Through Gradient Control : Stage Three

The third adaptation process focusses on the manipulation of the consequent sets. The process uses information from the target domain coupled with data produced from the framework itself.

This allows for feedback from within the adaptation framework. The process can be summarised as four steps:

Step One: Data Gathering

- A predefined n sized sliding window $\mathcal{SL}$ of data is collected from the source domain $\mathcal{D}_s$ and the target domain $\mathcal{D}_t$ for each input variable $x \in X$ and output $y \in Y$. The output value for the target domain is taken from the FuzzyTL system itself. The source output recorded from the labelled data provided. A sliding window is used to try to remove anomalous readings in the datasets, and to apply a form of smoothing. Gradients are formed based on the sliding window data between the input and the output value. This provides an understanding of the relationship at each data point. The gradients are the basis of the consequent adaptation.

Step Two: Gradient Production

- For each input and output in the source dataset, a gradient is produced. Similarly, for the target data set, gradients are formed for the input values. Output from the FuzzyTL framework is used to produce the target output gradient. The gradients are formed using a normalisation based on the standard score method. The standard score can be defined as:

$$z = \frac{x - \bar{x}}{\sigma} \tag{3.19}$$

  where z is the output, x is the input value, $\bar{x}$ is the mean of the sliding window and σ is the standard deviation of the sliding window. The standard score method allows for the comparison of values within differing domains through the use of the mean and standard deviation within the population used.

Step Three: Gradient Comparison

- Using the gradients gained across each source and target domain input and output variable, a comparison is made at each individual input value.

Step Four: Consequent Adaptation

- A mapping is made from the source input and output values, to the target input and output values based on the gradients of the values. By mapping the source gradient to the target gradient, differences can highlight the necessity to adapt the consequent sets. Differences between the source and target consequent gradients produce adaptations to the target consequent domain interval.

The adaptation of the consequent can be stated as:

$$d\mathcal{D}_a = \varphi \sum_{i=1}^{n} (g_{s_i} - g_{t_i}) \tag{3.20}$$

where φ is a learning parameter, g_s is the gradient of the source sliding window for n inputs that can be represented as

$$g_{s_{i \dots n}} \in [-1, 1]$$

,g_t is the gradient of the target sliding window for n inputs that can be represented as

$$g_{t_{i \dots n}} \in [-1, 1]$$

and $d\mathcal{D}_a$ is the delta used to adapt the consequent sets.

Figure 3.7 illustrates an example of the process described previously. Figure 3.7 shows a single input and single output taken over time. The sliding window used is taken across the same time interval for each variable. In the example given, time is measured in hours, light is measured in Lumens (lm), and temperature in degrees Celsius ($^{\circ}\mathrm{C}$). Gradients are produced by measuring the source input light against time over the defined interval. This is defined as S_i. Similarly, a gradient for the output is gained by measuring time against the temperature. In the same vein, a gradient is formed for the target task that stems from information in the FuzzyTL framework. As the target task is formed from unlabelled data, the output gradient is produced from information within the FuzzyTL framework process. This acts as feedback within the online learning process.

The gradients of the source and targets are compared. Where differences occur, adaptation is made to bring the consequent closer in line to the source data. The comparison is based on a number of rules.

In Figure 3.7 there is equality across the source and target input gradients, S_i and T_i. Both are positive. The output gradients, however show inequality. The source output gradient produces a negative value. Across the same time interval the target output gradient is positive. In order to alter this gradient, the domain of the target consequent sets are adapted. Positive differences between the output gradients produce a reduction in the domain. Inversely, negative differences produce positive movement. The quantity of the domain adaptation is based upon a weighted value. The weighting is formed using the difference between the source and task input gradients combined with the output domain interval. In order to adapt the system at a restricted rate, a weighting of these values is used. These are defined by the structure of the source data.

The initial three stages of the adaptation process approach the alteration of the fuzzy sets. In order to absorb the contextual changes within each implementation, the system additionally

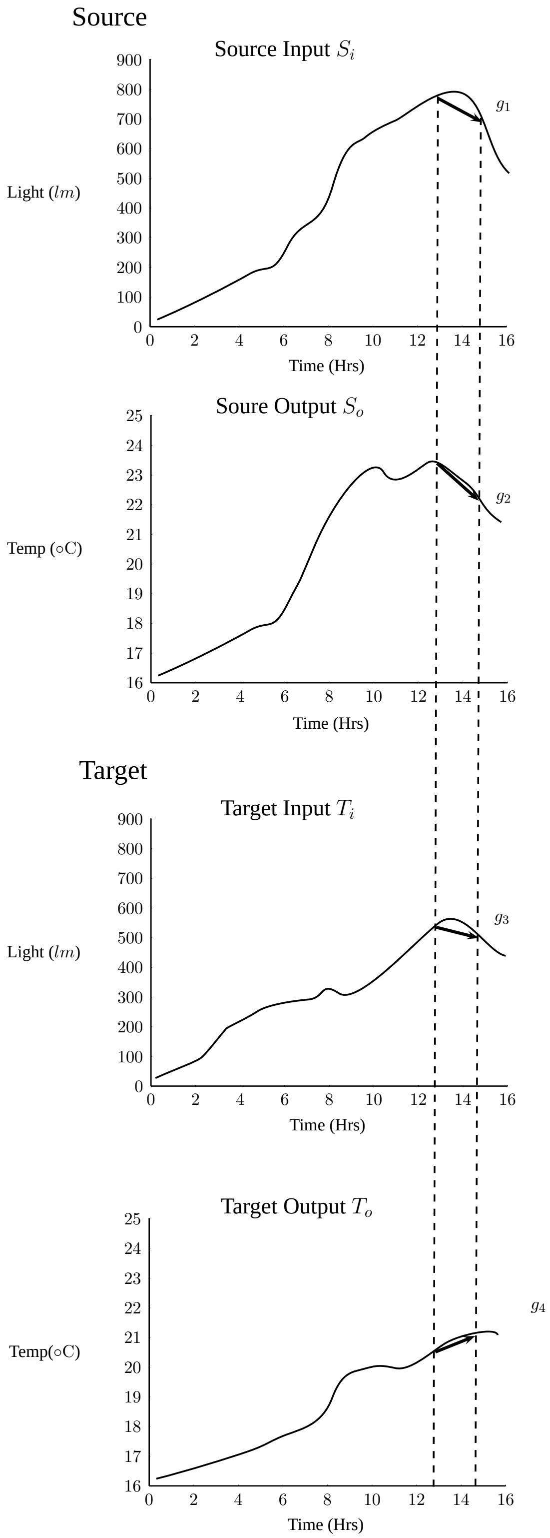

Figure. 3.7. Example of Gradient Analysis for Adaptation of Consequent Sets

adapts the fuzzy rulebase. Stages four and five address the issues that surround the adaptation of the rulebase by specifically focussing on the knowledge gap that occurs within the learning structure.

3.5.4 Rule Base Modification Via Source Rule Comparison : Stage Four

As differences can occur in the structure of the domains, equally differences can occur in the required learning process. Using the analogy previously introduced in Section 3.5.1, if the same individual decides to take up mountain biking, a portion of the skills developed road cycling are still applicable. The learning structure, however, has altered. New skills are required in order to accomplish the target task, although the inputs remain the same. These skills can come from new information, or an adaptation of previously acquired skills. Stages Four and Five approach the adaptation and restructuring of the learning gap through the use of previously acquired and newly formed information.

The knowledge of the FuzzyTL framework is held within the fuzzy sets and within the fuzzy rules. By altering the rulebase, the knowledge gaps can be filled in order to apply the transferable FIS to the new context.

In this stage, the rulebase is modified using the exhaustive rulebase created in the first stages of the framework. Rules that have been previously pruned are examined and applied to the target domain data to verify the applicability. Through an iterative process, the exhaustive rulebase is assessed to identify those rules that have a greater weighting, so greater applicability to the data within the target domain. The use of the exhaustive rulebase is firmly in keeping with the TL ethos of the frameworks construction. Through the use of the information contained within the source domain, the framework improves the ability of the target learning process through the use of previous knowledge. Algorithm 3.3 expresses the adaptation of the rules using the exhaustive transferable rulebase.

The first stage of the process is to examine the exhaustive rulebase $\mathcal{A}$ to identify any rules that fire using the data from the target domain $\mathcal{D}_t$. The rule that fires with the highest membership value from each data point is retained in the adaptive rulebase $\mathcal{C}$. The grouped rules are compared to the reduced rulebase $\mathcal{B}$ based on those with the same antecedent values. Each of the reduced rulebase rules that fires is compared to the adaptive rulebase. Those rules that have greater membership values are retained, removing the comparable rule from the adaptive rulebase. If the identified rule in the reduced rulebase $\mathcal{B}$ is not within the adaptive rulebase $\mathcal{C}$, this is added. The addition of the rules from the exhaustive rulebase assists in supplying missing knowledge areas required by the new task.

The final stage of the adaptation again focuses on the fuzzy rulebase.

Algorithm 3.3 Adaptation Algorithm: Stage Four Adaptation Using Exhaustive Rule Base.

```
Input Variable x ∈ X
Target Domain D_t = {(x_1^t, x_2^t)}_t^M
Exhaustive Rule Base A = {^f X_1^a, ^f X_2^a, ^f Y^a}^N
Reduced Rule Base B = {^f X_1^b, ^f X_2^b, ^f Y^b}^P
Adaptive Rule Base C = {^f X_1^c, ^f X_2^c, ^f Y^c}^Q
Membership Degree μ
Rule r = {^f X_1, ^f X_2, ^f Y}

for h = 1; h < M; h++
    for i = 1; i < N; i++
        e = μ_{^f_i X_1^a}(_h x_1^t) · μ_{^f_i X_2^a}(_h x_2^t)

        if e > 0 then                                  ▷ Check if the rule produces output.
            for u = 0 → Q; u ← i + u
                g = μ_{^f_u X_1^c}(_h x_1^t) · μ_{^f_u X_2^c}(_h x_2^t)

                if μ_{^f_i Y^a}(e) > μ_{^f_u Y^c}(g) then  ▷ Compare membership of A and C rules.
                    if C ≠ ∅ then
                        C −= C_k                       ▷ Remove current rule.
                    end if
                    C += A_i                           ▷ Add rule from exhaustive rulebase.
                end if
            end for
        end if

        for j = 1; j < P; j++
            for k = 1; k < Q; k++
                if (^f_j X_1^b == ^f_k X_1^c) ∧ (^f_j X_2^b == ^f_k X_2^c) ∧ (^f_j Y^b == ^f_k Y^c) then
                                                       ▷ Check if set labels are the same.
                    w = μ_{^f_u X_1^b}(_h x_1^t) · μ_{^f_u X_2^b}(_h x_2^t)
                    if μ_{^f_p Y^b}(w) > μ_{^f_k Y^c}(e) then
                        C −= C_k                       ▷ Remove current rule.
                        C += B_j                       ▷ Add rule from reduced rulebase.
                    end if
                end if
            end for
        end for
    end for
end for
```

3.5.5 Rule Adaptation Using Euclidean Distance Measure : Stage Five

Previously learnt information can provide data to partially fill gaps in the knowledge to complete a new task. To remove incompleteness, and strive to capture all of the segments where disparities lay, new information is required. In the FuzzyTL framework, information is partially embodied within the fuzzy rulebase. To reinforce the rulebase, new rules need to be constructed. As the task domain is an unlabelled dataset, this process relies on the use of the combined learning from the newly accumulated information and the use of previous knowledge in the form of the exhaustive rulebase. To produce the antecedent and consequent fuzzy sets, separate strategies are used.

The initial stage of the process is to gain an output from each of the input variables. The process can be demonstrated using a simple example. The domain is segmented into fuzzy sets, $(2N + 1)$. Within this example five sets are used. These are defined as $\{$ *Very Low, Low, Medium, High, Very High* $\}$. The construction of the sets is outlined in Section 2.3.5.1. The input value is applied to each set iteratively. The set with the highest output corresponds to the antecedent set for the new rule. This can be described as:

$$x_1 = \{< \text{Very Low}, 0.0 >< \text{Low}, 0.0 >, < \text{Medium}, 0.35 >, < \text{High}, 0.65 >, < \text{Very High}, 0.0 >\}$$

$$x_2 = \{< \text{Very Low}, 0.0 >< \text{Low}, 0.8 >, < \text{Medium}, 0.2 >, < \text{High}, 0.0 >, < \text{Very High}, 0.0 >\}$$

The example highlights two domains that are segmented into five sets. The domains represent the input variables x_1 and x_2 within the target task $\mathcal{D}_t$. The set with the highest firing strength produces the output for the antecedent sets of the rule. If sets of jointly strong firing strength are found, the initially discovered set is used. In the example shown, two antecedent sets relating to two input variables are formed from the highest memberships of the x_1 and x_2 domains. In the x_1 domain, the *High* set produces the highest membership. In x_2, highest is the set *Low*. As a result the *High* and *Low* are placed into the rule.

The formation of the antecedent sets is based on the domains formed by the previous stages. The adaptation that has occurred has endeavoured to move the input domains contextually towards their true state. As there is no availability of the consequent data, a different strategy is needed to construct the final element of the rule.

To produce the consequent set, a distance measure based on transferred data from the source domain is used. To initialise this measure an n dimensional euclidean distance based on the source input values against the target input values is calculated. Based on the closest overall value, the corresponding set identified within the exhaustive rulebase $\mathcal{E}_s$ is added to the previously formed antecedent sets.

During the formation of the exhaustive rulebase, each set is assigned a corresponding value.

This mapping is based on the value used to produce the output. Figure 3.8 shows the relationship between the source and target data values. From the source data, a mapping can be produced from the original input data values to the corresponding antecedent sets. Through the use of an n dimensional euclidean distance, the closet target input values can output sets based on these values. In the example, the sets are represented by the grid structure. Each square shows the relationship of the input values, and the corresponding sets. Shown in Figure 3.8 are two target antecedent values, t_1 and t_2. These values are not represented within the current reduced rulebase. By mapping them to source input values, the antecedent sets can be found. Using the smallest euclidean value, Figure 3.8 shows that t_1 can be mapped to s_1 via the distance d_1. Equally, t_2 can be mapped to s_2 via the distance d_4. Through this procedure, a combination of antecedent sets is formed. Using these sets as a comparative value, a consequent set can be extracted from the exhaustive rulebase. By extracting the consequent set in this manner, a new rule is formed that draws knowledge from the source dataset.

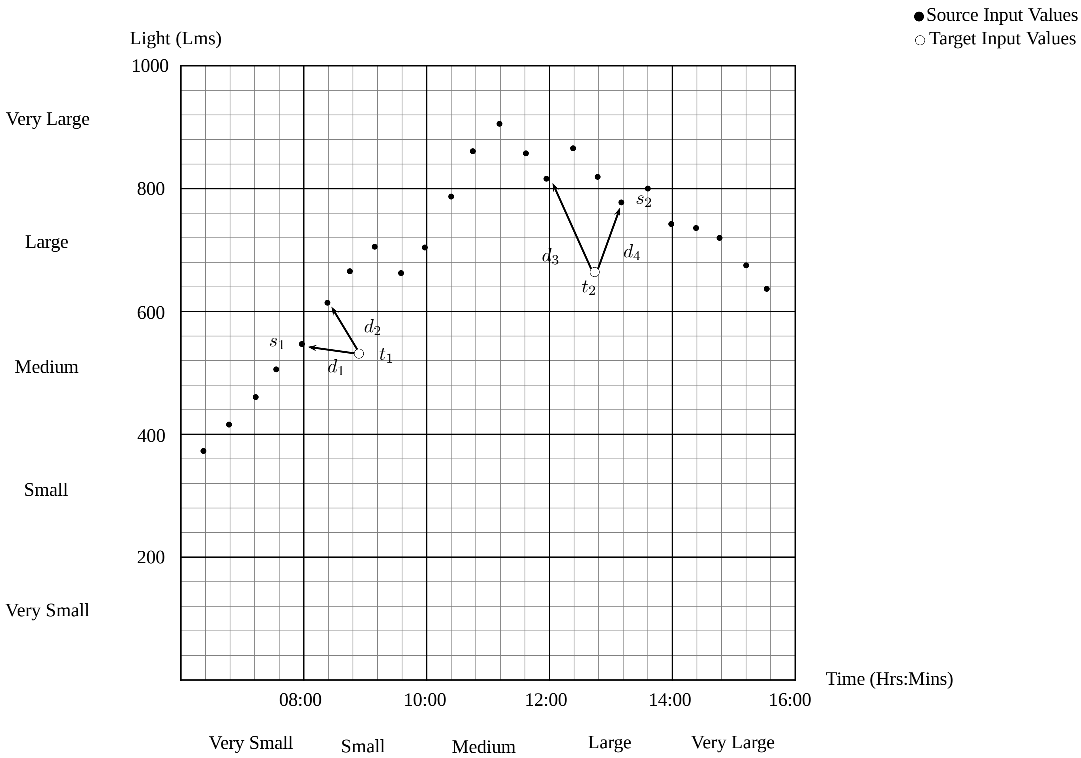

Figure. 3.8. Example of the Gaining of Antecedent Sets to Map Consequent Sets Using Euclidean Distance.

Taking the example shown in Figure 3.8 the consequent production of the rule is carried out through the following process.

1. Any data instance taken into the the FuzzyTL framework that does not result in a rule being

fired requires a new rule to be produced. The initial step is to produce the antecedent elements. This has been described in the previous section. It can be assumed that from this process two antecedent values have been defined: Input Variable x_1 $=$ *High* , Input Variable $x_2 =$ *Low*. Using the example in Figure 3.8, the input variables would equal Time $=$ *High*, Light $=$ *Low*.

2. The consequent value is formed by comparing the data from the target task to data within the source. The mapped source input values xs are compared to the target values xt using a Euclidean distance. This can be described as

$$d(xs, xt) = \sqrt{\sum_{i=1}^{n}(xs_n - xt_n)^2} \tag{3.21}$$

   Based on the lowest distance value dl, the corresponding source input values xs are used to map to the consequent set c within the exhaustive rulebase $\mathcal{E}$. The source consequent set can then be added to the antecedent values previously produced and added to the rulebase.

3.6 Summary

In this chapter, the major elements of the research were presented. An overview of the Fuzzy Transfer Learning (FuzzyTL) methodology was discussed, with a detailed view of its constituent parts. The major elements of the framework presented in this chapter can be summarised in two sections:

1. The transfer and reuse of fuzzy sets and rules combined with the application of a fuzzy frequency extension to the Wang-Mendel (WM) rule pruning methodology.

2. The implementation of a five stage online learning methodology to adapt contextually different information to produce output for target tasks.

The elements that are contained within this chapter are fundamental to the understanding of the following chapters. The experimental work that is carried out in Chapter 4 uses the framework outlined here.

3.6.1 Summary of Transferring Fuzzy Concepts

The initial section of this chapter proposed the concept of the transferral of fuzzy elements as a learning base. Using the WM rule extraction process, a fuzzy frequency rule pruning process is set out as an addition to this methodology. This process is a novel addition to the WM methodology. The fuzzy frequency pruning method looks to remove issues produced by over-represented anomalous data within the production of rules. As a result, this process may have further application outside of the scope of this framework.

3.6.2 Summary of Adaptation Through Learning

A major element of the FuzzyTL framework is the use of an online adaptive learning process. Drawing on the foundations of Transfer Learning (TL) and similar areas such as domain adaptation, a five stage adaptation process was presented in this chapter. The first stage uses newly sourced information from the target environment to adapt the target domain. This process addresses issues that are produced through the differences in data between the source and target tasks. The adaptation of the domain alters the structure of the fuzzy sets, allowing the changes to be represented in the Fuzzy Inference System (FIS).

Stage two also focuses on the adaptation of the fuzzy input set structure. The secondary stage adapts the sets based on combined information from the source and target tasks.

The third stage focuses on the consequent sets within the FIS. Using fundamental aspects of TL, gradient information from the source task is transferred to the target learning task. Differences between the source and target gradient relationships provide parameters to adapt the consequent sets.

The final two stages depart from the adaptation of fuzzy sets. These stages confront issues that arise within the fuzzy rulebase during the transfer of an FIS across contextually differing situations. To reduce these issues, stage four uses the exhaustive rulebase to extract rules more focussed on the new target task. Through the use of a comparative process, the rulebase is expanded, adjusted or pruned to increase the output firing strength.

The transfer of a source rulebase can result in no rule matching the target data. Stage five addresses this issue focussing on completeness in the rulebase. New rules are produced through a method of antecedent set extraction, coupled with the production of consequent sets using a euclidean distance approximation. By filling the information gap that is produced by the transferral process, stage five allows the framework to output values across all data inputs.

The following chapter presents an implementation of the FuzzyTL framework in an Intelligent Environment (IE) application across two differing data sets gained in two alternate temporal and contextual situations.

Chapter 4

Fuzzy Transfer Learning in Intelligent Environments

4.1 Introduction

Intelligent Environments (IEs) are complex and dynamic. The data produced can vary significantly across implementations and from context to context. To help illustrate the performance of the Fuzzy Transfer Learning (FuzzyTL) framework in changing contextual situations, this chapter implements the framework using data gathered across two separate IEs. The results, and subsequent analysis answer the hypotheses outlined in Chapter 1.

The chapter is set out as follows: Section 4.2 gives an in depth view of the design of the experiments including the datasets produced, and the methods used to construct them. The following sections, Sections 4.3, 4.4 and 4.5 will discuss the results gained from the experimentation results with the final section, Section 4.6, offering a summary of the findings.

4.2 Experimental Design

The motivation for the FuzzyTL framework is to address the issue of environments where little or no knowledge is known *a-priori*, though there is a need to produce a prediction or classify the target data. Torrey and Shavlik (Torrey & Shavlik 2009), as discussed within Chapter 2 Section 2.4 suggest three metrics that can be used to measure the performance of a Transfer Learning (TL) system. To recap, these are defined as:

1. The initial performance achievable in the target task using only the transferred knowledge, before any further learning is done, compared to the initial performance of an ignorant agent.

2. The amount of time it takes to fully learn the target task given the transferred knowledge compared to the amount of time to learn it from scratch.

3. The final performance level achievable in the target task compared to the final level without transfer.

These metrics are closely associated with the construction of learning processes, and the composition and quantity of the available data. The metrics proposed by Torrey and Shavlik require the TL structure to be Informed Supervised (IS), where information is available from both the source and target domains. Informed Unsupervised (IU) transfer learning, as used in this thesis, makes the comparison of ignorant and informed agents not possible. To compare the performance of a starting ignorant agent is not feasible, as there is no information to model the agent upon. To learn an agent from scratch also requires a level of known data. This research approaches a problem where there is no labelled data within the target domain, and initially little or no unlabelled data. For this reason, the second metric proposed by Torrey and Shavlik is not applicable. For this reason, a different set of metrics are used, however, they are broadly based on the those proposed in (Torrey & Shavlik 2009).

The metrics use a comparison of the output of FuzzyTL framework, against actual known sensor readings from the IEs. Input values are given to the system producing a predictive value. This is compared to actual recorded sensor readings. The error indicates the accuracy of the FuzzyTL system.

In a real-world scenario, this comparison could not be accomplished. Availability of labelled data would alter the learning scenario from IU, to an IS transfer process. This is often the ideal situation. Labelled target data increases the amount of information available to the learning process. Complete labelled data of the target task provides ground truth of the problem. This can be used as a comparison against scenarios where less information is available.

As defined in Chapter 3, labelled data takes the form of $D_l = \{(x_1, x_2, y)\}^N \in \{X_1, X_2, Y\}^N$ where D is the dataset, $x \in X$ is an input variable, $y \in Y$ is an output variable and N is the number of data tuples in the dataset. Unlabelled data is expressed as $D_u = \{(x_1, x_2)\}^M$ where m is the number of data tuples in the dataset. For this comparison, the target dataset D_t is defined as $D_t = \{X_1^t, X_2^t, Y^t\}^N$, though the output Y^t is only used for comparative purposes.

To evaluate the use of the FuzzyTL framework, two real world Intelligent Environment (IE) datasets were chosen to demonstrate the applicability of the spatial and temporal contextual transfer process. Predominantly IEs are constructed using a large number of varying types of sensor ranging from temperature and humidity sensors within environmental monitoring (Jung et al. 2008) to Passive Infra-red Sensor (PIR Sensor) within smart home structures. The implementations of such networks result in a wide array of dynamic data sources. The quantities of sensors used, possibly in excess of 100pcs in a single deployment, can produce large quantities of data, and the uncertain and dynamic form of the environments make the construction of models extremely difficult. IEs offer dynamic and uncertain data production that are prime examples of the scenarios that the FuzzyTL framework can address.

The first dataset was taken from a publicly available source. Details to access this data can be found in the bibliography at (Madden 2004).

4.2.1 Intel Berkeley Research Laboratory Dataset

The Intel dataset is based upon information collected from 54 sensors deployed in the Intel Berkeley Research Laboratory (hereby referred to as the Intel Laboratory) between the 25th February and the 5th April, 2004. The network used XBow Mica2dot weatherboard based nodes to record environmental data across the internal structure of the laboratory. Four parameters were measured: time-stamped temperature (in degrees Celsius), humidity ranging from 0-100%, light (measured in Lux), and residual power of each sensor unit expressed in Volts. Figure 4.1 depicts a single Mica2dot sensor.

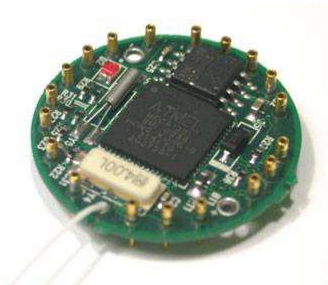

Figure. 4.1. Crossbox Mica2bot Wireless Sensor (*The Sensor Network Museum* 2012).

The data was collected using the TinyDB in-network query processing system built onto the TinyOS platform which recorded information every 31 seconds (Madden 2004). The layout of the nodes can be seen in Fig.4.2.

An example of the data produced is shown in Table 4.1. The raw data includes the date in the year-month-day format, and the time in the hour-min-sec.millisecond format. The variables temperature, humidity, light and voltage are recorded as real numbers. Additionally to these values, the TinyDB system records both an epoch and a moteid. The epoch is a monotonically increasing sequence number from each sensor (or mote). The same value can be produced from different sensors at the same time. The moteid is a unique identifier for each sensor. This ranges from 1-54. These values can be seen on Figure 4.2. The nature of the construction of the sensor network and the real world application resulted in data missing from individual sensors across certain time periods or being truncated.

Additional to the sensor readings, the Intel Laboratory provides the co-ordinates of the sensor locations. These are X and Y co-ordinates, relative to a single point of the room (as depicted

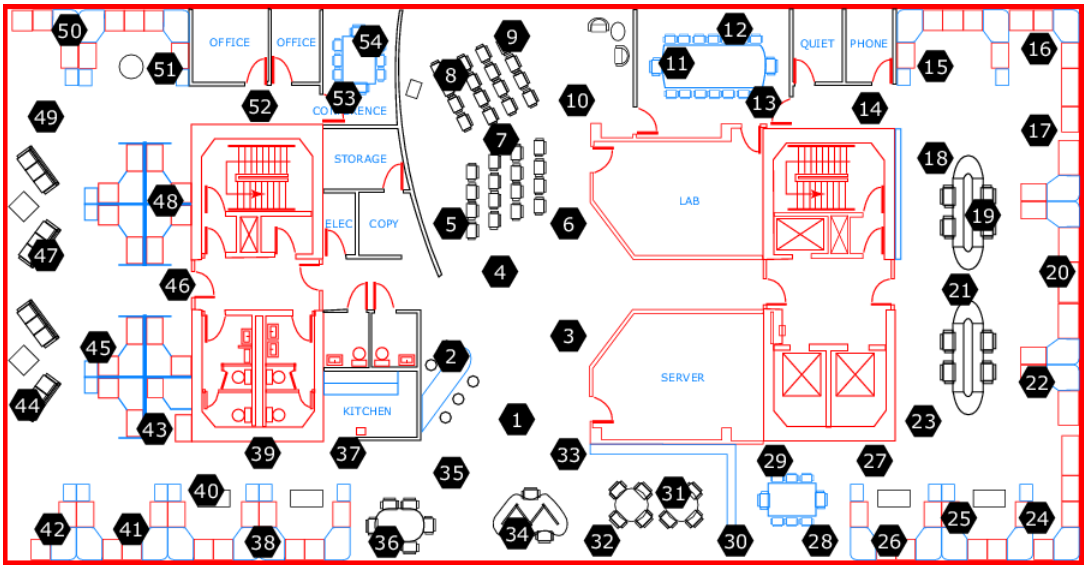

Figure. 4.2. Diagram of Intel Laboratory Showing Placement of Wireless Sensor Nodes From (Madden 2004).

by the upper right hand corner of Figure 4.2). These values were used in the experiments to construct a context measure. A section of network was identified across the laboratory to examine the influence of variations in the spatial aspect of the contexts. Data was also taken from a section of the dataset that related to a specific time period. This, in turn, allows for the investigation of the effect of temporal changes to the context. To achieve both of these, the output of Sensors 7, 9, 12 , 24 , 34 , 42 and 51 where examined across seven days from 28th February to 5th March, 2004 including the 29th February. The locations of the sensors are given in Table 4.2.

A quantity of preprocessing was undertaken to be able to place the dataset into the FuzzyTL framework. Each sensor was isolated based on its moteid. The unused variables were also removed from the data resulting in only the time, light and temperature remaining. The time variable was converted to seconds to allow for ease of processing. The millisecond component was removed allowing for this process. A number of different experimental set ups were used to illustrate the effect of greater and lesser availability of labelled data in the source domain. These scenarios will be discussed further in Section 4.4.1.

4.2.2 De Montfort University Robotics Laboratory Dataset

The second dataset is based on a sensor network constructed in the Robotics Laboratory of the Centre for Computational Intelligence of De Montfort University, United Kingdom. Again the sensor network is focussed on the monitoring of environmental conditions. The Robotics

Date:	Time:	Epoch:	Moteid:	Temperature:	Humidity:	Light:	Voltage:
2004-02-28	00:59:16.02785	3	1	19.9884	37.0933	45.08	2.69964
2004-02-28	01:03:16.33393	11	1	19.3024	38.4629	45.08	2.68742
2004-02-28	01:06:16.013453	17	1	19.1652	38.8039	45.08	2.68742
2004-02-28	01:06:46.778088	18	1	19.175	38.8379	45.08	2.69964
2004-02-28	01:08:45.992524	22	1	19.1456	38.9401	45.08	2.68742
2004-02-28	01:09:22.323858	23	1	19.1652	38.872	45.08	2.68742
2004-02-28	01:09:46.109598	24	1	19.1652	38.8039	45.08	2.68742
2004-02-28	01:10:16.6789	25	1	19.1456	38.8379	45.08	2.69964
2004-02-28	01:10:46.250524	26	1	19.1456	38.872	45.08	2.68742
2004-02-28	01:11:46.941288	28	1	19.1456	38.9401	45.08	2.69964
2004-02-28	01:12:46.251377	30	1	19.1358	38.9061	45.08	2.68742
2004-02-28	01:14:16.63127	33	1	19.1162	38.8039	45.08	2.69964
2004-02-28	01:14:46.569352	34	1	19.1162	38.872	45.08	2.69964
2004-02-28	01:15:16.649556	35	1	19.1064	39.0082	45.08	2.69964
2004-02-28	01:16:16.343708	37	1	19.1064	38.872	43.24	2.69964
2004-02-28	01:16:46.508622	38	1	19.0966	38.8039	43.24	2.69964
2004-02-28	01:17:46.42744	40	1	19.0966	38.7357	43.24	2.69964
2004-02-28	01:18:16.468248	41	1	19.0868	38.8039	43.24	2.69964

Table 4.1. Example of Intel Laboratory Dataset Structure.

MoteID	Position X (m)	Position Y (m)
7	22.5	8
9	21.5	2
12	13.5	1
24	1.5	30
34	21.5	30
42	39.5	30
51	35.5	4

Table 4.2. Position of Sensors Used in Intel Laboratory Dataset.

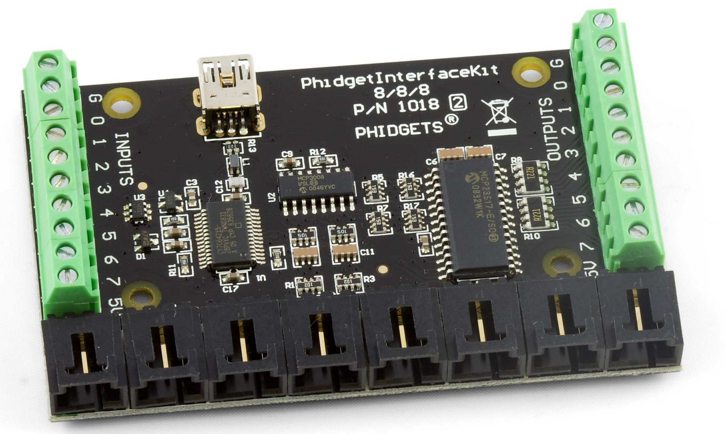

Figure. 4.3. Phidget USB Interface 8/8/8.

Laboratory sensor network is based around a Phidget architecture using Phidget USB interface boards at its heart. The boards have a Universal Serial Bus (USB) used to connect directly to a computer. The boards used can be seen in Figure 4.3. The Phidget Interface boards offer analogue inputs that can be used to measure continuous quantities such as temperature, humidity or position. The sample rate was set to 30 seconds.

The De Montfort University Robotics Laboratory Sensor Network (here after referred to as the Robotics Laboratory) was composed of nine sensors in total. The network centred around two Phidget 8/8/8 boards, each with Phidget temperature and light sensors. Combined with this were binary switches to monitor the opening and closing of both windows and doors within the room. The light and temperature sensors were housed in the same location within the laboratory. Figure 4.4 shows the structure of the Robotics Laboratory.

Sensors 3, 5 and 6 (represented as Sn3, Sn5 and Sn6) are composed of Phidget light and temperature sensors. Sensors 1, 2 and 4 (Sn1, Sn2 and Sn4) are single temperature sensors.

The sensor can measure ambient light up to 1000 Lux which is approximately the equivalent of a typically lit television studio or overcast day. The sensor requires no calibration as this is predetermined in the factory. The Phidget Temperature sensor has a range of $-30°\mathrm{C}$ to $+80°\mathrm{C}$. In the range $0°\mathrm{C}$ to $80°\mathrm{C}$, the sensor produces a typical error of $0.75°\mathrm{C}$.

The structure of the network follows a simple star design. Each sensor is connected to a Phidget board, which is subsequently connected to a PC. Five days of data were collected between the 15th and 19th October, 2011. The data was synchronised using the network time applied to each of the PC's that the boards were connected to. This removed issues that can be encountered

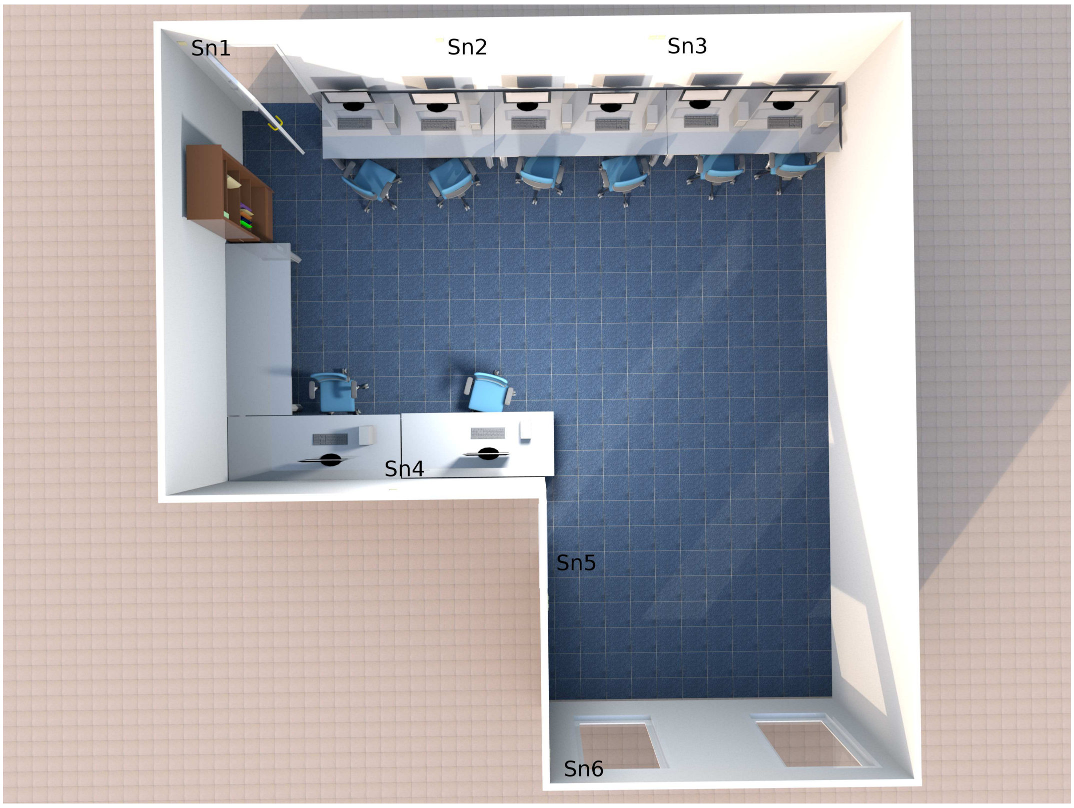

Figure. 4.4. Top Down View of the Structure of the Robotics Laboratory Highlighting the Sensor Network.

through differing time / date tags.

In the same vein as the Intel Laboratory dataset, the time, light and temperature were isolated for each sensor. Three of the sensors were used in order to apply the input variables of time and light, and an output variable of temperature. These sensors were Sn3, Sn5 and Sn6. All superfluous information was removed from the raw data leaving each sensor, and the day of the week isolated. The locations of the sensors is shown in Table 4.3. Each position relates to an x and y location being relative to the front right hand corner of the room (see Figure 4.4).

Sensor	Position X (m)	Position Y (m)
3	5.43	3.06
5	3.79	1.87
6	3.79	0.13

Table 4.3. Position of Sensors Used in Robotics Laboratory Data Set.

The construction of the framework to process both the Intel Laboratory and Robotics Laboratory datasets were constructed and run using C++ via Code:Blocks (Version 8.02) and compiled through GNU GCC on Ubuntu LTS Version 10.04.

4.2.3 Experiment Structure

The two hypotheses that will be tested are restated below.

Hypothesis 1: Where minimal unlabelled data is available within a target task, data in the form of a TL process from contextually related but differing source tasks, can be used to learn predictive tasks.

Hypothesis 2: Adaptation of the transferred source domain through the use of unlabelled new data can increase the performance FuzzyTL in predicting target tasks.

Hypothesis 1 will be evaluated primarily through the use of the *Performance* and *Context Impact* experiments. Hypothesis 2 will be evaluated using the *Adaptation* experiments.

To address the hypotheses set out in Chapter 1, three main experimental groups were carried out. These can be categorised under broad headings.

1. *Performance:* To evaluate the overall performance of the FuzzyTL framework, the system was used to calculate an output value in the target domain based on an increasing quantity of unlabelled data. The initial quantity is zero indicating no prior knowledge of the target domain. The output from the FuzzyTL framework was compared against the actual reading recorded by each sensor network. A Root Mean Squared Error (RMSE) of the FuzzyTL

framework output and the actual sensor output was used to evaluate the performance of the predicted value.

2. *Context Impact:* To assess the impact of inter contextual differences, the source data was assessed against the target data based on a contextual distance metric. The metric is described further in the following section. Correlation between contextual distance and performance was investigated. The impact of the use of intra context information was assessed through a performance comparison of source data from both datasets.

3. *Adaptation:* To understand the necessity for the use of adaptation in the learning structure within the FuzzyTL framework, the performance of non-adapted, transferred systems was compared to the performance of the full FuzzyTL system. The same evaluation criteria as the context and performance were used.

Using the experimentation, the hypotheses were answered by stating that:

- The performance of a *best* output sample of the FuzzyTL framework was comparable to a benchmark sample dataset. This showed the capability of the framework to use contextually related but different data to predict tasks confirming hypothesis 1.

- A comparison using both the Intel Laboratory and Robotics Laboratory datasets showed a reduction in RMSE when the five stage adaptive processes were applied. This confirmed the second hypothesis.

Definitions This section will define a number of elements used through out the remainder of this chapter. To calculate the difference between the predicted value produced by the FuzzyTL framework and the actual values that are observed, a RMSE is used. The RMSE takes the errors between each of the points in the dataset, and aggregates them into a single measure. RMSE can be defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_1^i - x_2^i)^2}{n}} \tag{4.1}$$

were n is the number of data points in the dataset, $x1$ is the observed dataset and $x2$ is the predicted value.

To understand the impact of the contextual change, a metric was produced to measure the temporal-spatial difference between context structure. A normalised euclidean distance was used. Three separate inputs were given, the specification of the sensor location using the x and y co-ordinate, and a date measurement. This is constructed as

$$CD = \sqrt{\frac{((a^1 - a^2) - \sup A)}{\inf A - \sup A}^2 + \frac{((b^1 - b^2) - \sup B)}{\inf B - \sup B}^2 + \frac{((c^1 - c^2) - \sup C)}{\inf C - \sup C}^2} \tag{4.2}$$

were a is the x and b is the y co-ordinate, c is the date, and CD is the Context Distance (CD). The $\sup$ and $\inf$ are calculated to produce a normalised distance. This allows for different values to be used. A and B define the spatial inputs of the context. C is the temporal input. The context can be composed of n inputs. The context used in these experiments consists of spatial and temporal elements. Different contexts may contain different variables. For example, two Wireless Sensor Networks (WSN's) are used to monitor growth patterns and the effect of the environmental conditions of forests in Scotland and North America. Each Scottish context is defined using three dimensions (x,y,z), time and date, and the proximity of the sensors to a single natural feature of interest (a river). The North American Wireless Sensor Network (WSN) is composed of two dimensions (x,y), time and date, and the proximity of the sensors to a natural feature of interest (a mountain). Both WSN's also record light levels, rain fall and wind speed.

To study context in relation to the FuzzyTL framework, two differing datasets were chosen. These have been outlined in Sections 4.2.1 and 4.2.2. Context has been defined previously in Chapter 2 as consisting of three elements:

Information Each context consists of definable variables that are relevant and measurable.

Behaviour The context embodies an entity, application, service or group thereof that is affected by the behaviour of the associated information.

Variation Differences within the structure of the variables can occur between context to context, but not from instance to instance within a context itself. This would be defined as a new context.

Based on this criteria, two further sub-types of context will be used within this chapter, *Inter* and *Intra* contexts. The concepts of inter and intra contexts can be illustrated by expanding the example used previously. Focussing on the Scottish WSN, an *inter* context can be defined for a section of trees that encompass a geographical area. Individual contexts can subsequently be defined for single trees, or individual days, weeks or months. Spatial and temporal comparison of distances between the individual trees or groups thereof can be made to ascertain the impact of geographical position or changing seasonal conditions. These are inter contextual comparisons. To compare the Scottish WSN to the North American, a categorical definition is given forming an *intra* context. This can simply be shown as *Scotland* and *America*.

Inter contextual comparison specifies the contexts as existing within a predefined scenario. This can itself be a location such as a building, a time scale or a composition of these. Within this thesis, an inter contextual comparison is composed within the individual locations of the sensor networks. Each context is then defined by time and location of the sensors within this location. This falls in line with the third criteria of a context. An intra contextual comparison uses an abstract definition to compare the contexts. Unlike the inter comparison, a categorical definition is given. Within this thesis, these context definitions are used for comparison. Each comparison takes the form of assessing the performance of the FuzzyTL framework based on differences in the individually defined contexts. The difference is calculated based on the defined variables that constituent the context. This allows for contexts with behavioural differences to be compared.

Within the following sections, further explanation of the construction of each individual experiment will be given, with the results that were gained. This will be followed by a summary focusing on the relevance to the hypotheses depicted in Chapter 1.

4.3 Performance

In this section, a number of experiments will be used to test the first hypothesis as set out in Chapter 1. To measure the performance of the FuzzyTL framework, two datasets were used. Across both datasets, a system was constructed using two input variables (time and light), and a single output variable, temperature. For this performance measure, inter contextual comparisons were produced. The initial value output of the system was based on zero prior unlabelled data. Each data point from the dataset was fed into the system to simulate real time operation. The adaptation of the FuzzyTL was based on an iterative increase of data. Full knowledge of the target domain inputs only occurred on completion of the data throughput. To assess the performance of the FuzzyTL framework the predicted value at each data point was compared to the actual observed output from the dataset. Any error produced was consolidated into a single value using the RMSE process.

The methodology described in Chapter 3 proposes the production of predictive outputs for target tasks were little or no training data is available, and where the training data is unlabelled. The scarce nature of the training data does not allow for standard learning approaches. The format of the data is additionally very restrictive. The unlabelled nature allows for only unsupervised learning to occur. In the initial stages the data is extremely sparse in quantity, reducing the ability for unsupervised learning to adequately function. The main focus of this thesis is the presentation of a novel method to produce output when such data structures occur.

To produce a benchmark to compare the system against, each of the datasets were processed using the adapted Fuzzy Frequency Wang-Mendel (WM) system as presented in Chapter 3, Section 3.4.2. The adaptation stages of the FuzzyTL were removed from the process. The learning process was, however, altered. The source data was supplied from the target domain. This produced an output that, unlike the FuzzyTL, has labelled knowledge of the target learning task. This allows a comparison to be made. The FuzzyTL framework that is supplied target data is perceived to produce an output closest to the actual sensor value. A comparison was made to demonstrate the context of the output from the contextually different source FuzzyTL framework.

4.3.1 Intel Laboratory Data Comparison to Observed Values

For the Intel Laboratory dataset, individual contexts were formed using each of the sensors in the spatial grouping (Sensors 7, 9, 12 , 24 , 34 , 42 and 51), and for each day between 28th February to 5th March, 2004. Extraneous source and target contexts were removed from the experiment. A single context is formed from a single day and a single sensor. A value was predicted for a single sensor taken from within the group, across the defined time period. This produced 2352 differing context comparisons. In the event that the system is unable to predict a reading, a predefined value of -1 is given.

To assess the performance of the FuzzyTL framework, source contexts were compared to the benchmark values. This provides an insight into the ability of different source data to provide the initial starting learning point for target predictive tasks. The adapted WM method produced 49 separate RMSE values (seven sensors $\times$ seven days). These related to each sensor (7, 9, 12 , 24 , 34 , 42 and 51), and each day within the specified time interval (from 28th February to 5th March, 2004 including the 29th February). 2352 contexts were produced for the Intel Laboratory comparison. These were composed of the 49 separate contexts from the source data (seven sensors $\times$ seven days) and the target data (seven sensors $\times$ seven days) with 49 contexts removed where the source and target context matched. From the 2352 RMSE values calculated from the FuzzyTL framework, the lowest RMSE value was taken for each context. This produced 49 contexts to compare to the benchmark set. The RMSE value was the comparison of the output from the framework and the actual real-world sensors. By isolating the lowest RMSE values, this equated to the *best* performing contexts. As the benchmark dataset represents the optimum data conditions for the learning process, the best performing contexts were chosen to compare against them. The focus of this process is to establish whether the FuzzyTL framework can firstly output a predictive value, and to then contextualise the performance.

The data for the experiment is composed of two sample population datasets, the benchmark and Intel Laboratory datasets. To compare the benchmark of the Intel Laboratory Data and the output of the FuzzyTL, the datasets were firstly tested for normality. Anderson-Darling tests were used to ascertain if the datasets were normal. A null hypothesis (H_0) describing the data as being normally distributed was set. The benchmark dataset was found to be not normal. Table 4.4 shows the results of this test. To normalise the data, a power transform was used on both datasets. A power transform takes the form of raising each value x to a power q. This took the form of $log_6(x)$. A second Anderson-Darling test was carried out to check for normality. Tables 4.5 and 4.6 show the results of the tests. In both datasets an alpha value α of 0.05 was used. This indicates the predetermined significance level of the test. The α level determines the probability to which the result occurred due to chance. A level of 0.05 gives 95% certainty in the result. The Anderson-Darling is a one-sided test requiring the p-Value to be greater than the α to reject the hypothesis. As the p-value calculated was higher than the α value, the dataset is viewed to have no significant departure from normality.

Anderson-Darling Test	
Data: Benchmark Intel Laboratory	
Alpha-value	0.05
P-value	0.0018
Observation Number	49
Conclusion	Not Normal

Table 4.4. Anderson-Darling Test Results For the Benchmark Intel Laboratory Output.

Anderson-Darling Test	
Data: Benchmark Intel Laboratory	
Alpha-value	0.05
P-value	0.0879
Observation Number	49
Conclusion	Possibly Normal

Table 4.5. Anderson-Darling Test Results For the Benchmark Intel Laboratory Output Using $log_6(x)$.

Anderson-Darling Test	
Data: Best FuzzyTL Intel Laboratory	
Alpha-value	0.05
P-value	0.9732
Observation Number	49
Conclusion	Possibly Normal

Table 4.6. Anderson-Darling Test Results For the Best FuzzyTL Intel Laboratory Output Using $log_6(x)$.

Further examination of the datasets show that the two medians and the 1st and 3 rd quantiles are in close proximity. Greater variance comes at the extremes of the datasets. This is shown in Figure 4.5.

To compare the two sample populations of the data, a paired t-test was used. This test was chosen as both sets of values were related. The use of the paired t-test was to retain or reject the null hypothesis H_0 that the benchmark Intel Laboratory and the Intel FuzzyTL framework were identical populations. An alpha (α) value of 0.05 was set for the test. As with the Anderson-

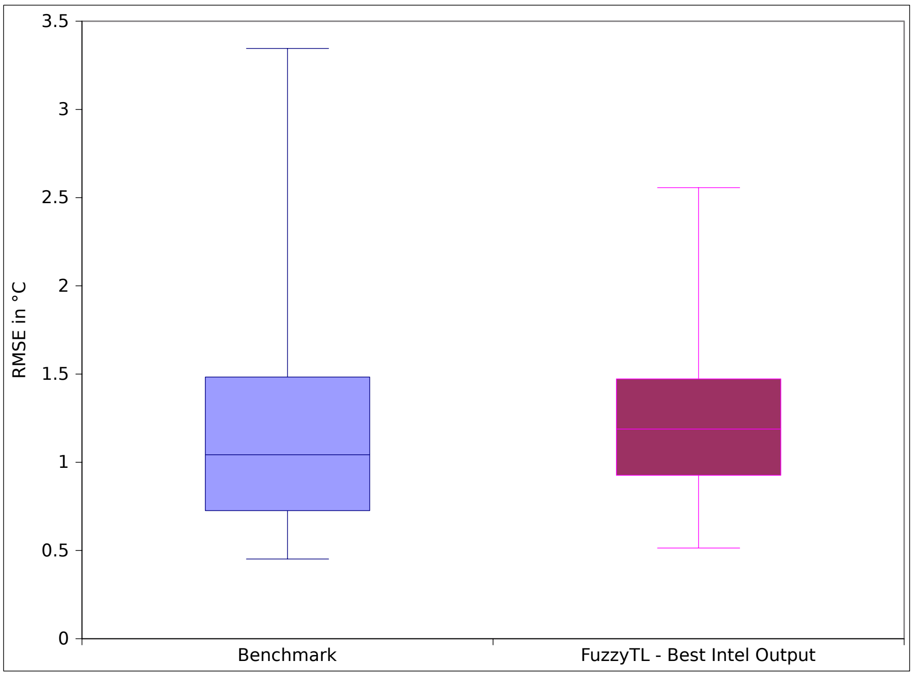

Figure. 4.5. Comparison of Adapted Wang-Mendel Benchmark and Best FuzzyTL Output Using Intel Laboratory Dataset.

	Intel Benchmark	Intel Best FuzzyTL
Min	0.4517	0.5138
Max	3.3455	2.5567
Median	1.0427	1.1883
Quantile 25%	0.726	0.9261
Quantile 75%	1.4832	1.4727

Table 4.7. Data From Comparison of Adapted Wang-Mendel Benchmark and Best FuzzyTL Output Using Intel Laboratory Dataset.

Darling test, a level of 0.05 gives 95% certainty in the result. The Table 4.8 shows the values calculated. As the p-value is below the α level ($0.0015 \leq 0.05$), it can be concluded that

Paired T-Test	
Data: Intel Laboratory Benchmark Compared to Intel Laboratory FuzzyTL	
df	48
T-Statistic	-3.7191
Alpha-value	0.05
P-value	0.0003
Positive Differences	37
Negative Differences	12

Table 4.8. Paired T-Test Results For the Intel Laboratory Benchmark and Best FuzzyTL Output.

the benchmark and FuzzyTL are from non-identical populations. There is significant difference between the datasets, so the null hypothesis is rejected. Looking closer at the data, 37 contexts are highlighted as producing a negative difference. This showed that the best performing FuzzyTL output was greater in 37 contexts than the benchmark. However, 12 of the contexts, 24.4898%, were a lower RMSE than the benchmark. In these cases the FuzzyTL system was able to use contextually different source data to produce better performing output than the benchmark dataset. Figure 4.6 shows a comparison of each of the benchmark contexts against the best values produced by the FuzzyTL framework.

Figure 4.7 shows a single context comparison where the best FuzzyTL output out performed the benchmark dataset. This is the source data 24, 28th February, 2004, and the target data Sensor 34, 28th February, 2004.

The initial close proximity of the input and output interval domains allowed the source to provide a good starting learning point. The interval values can be seen in Table 4.9.

	t_L	t_R	l_L	l_R	tm_L	tm_L
Target Sensor	3525	86391	57.03	1847.37	16.69	26.92
Initial Best FuzzyTL	3525	86391	60.72	1847.37	16.53	26.28
Adapted Best FuzzyTL	3525	86391	57.04	1847.37	16.51	26.26

Table 4.9. Comparison of Initial Input and Output Interval Domains For the Target Context Sensor 34, 28th February, 2004

The input domain intervals are moved closer to the target sensor values from the initial point. The ability for the framework to adapt the sets according to new data improved the output beyond the benchmark. The table shows the final consequent domain interval being further from the target sensor. This value is dynamic, however. The consequent is adapted based on the changing data input. The ability to adapt the consequent domain interval based on the data input allowed the

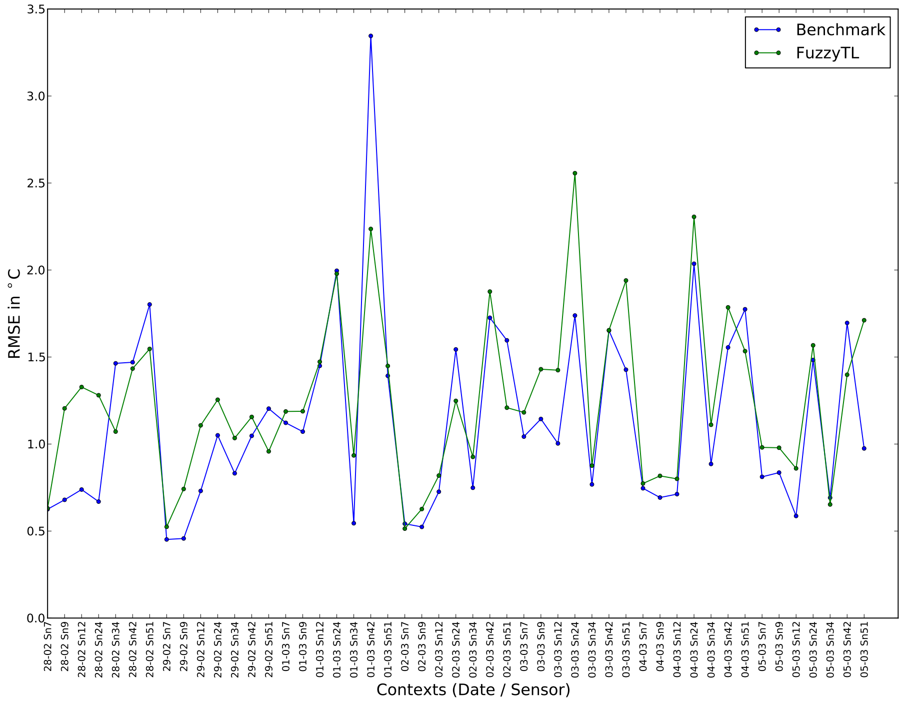

Figure. 4.6. Direct Comparison of Adapted Wang-Mendel Benchmark and Lowest FuzzyTL Output For Each Context Within the Intel Dataset.

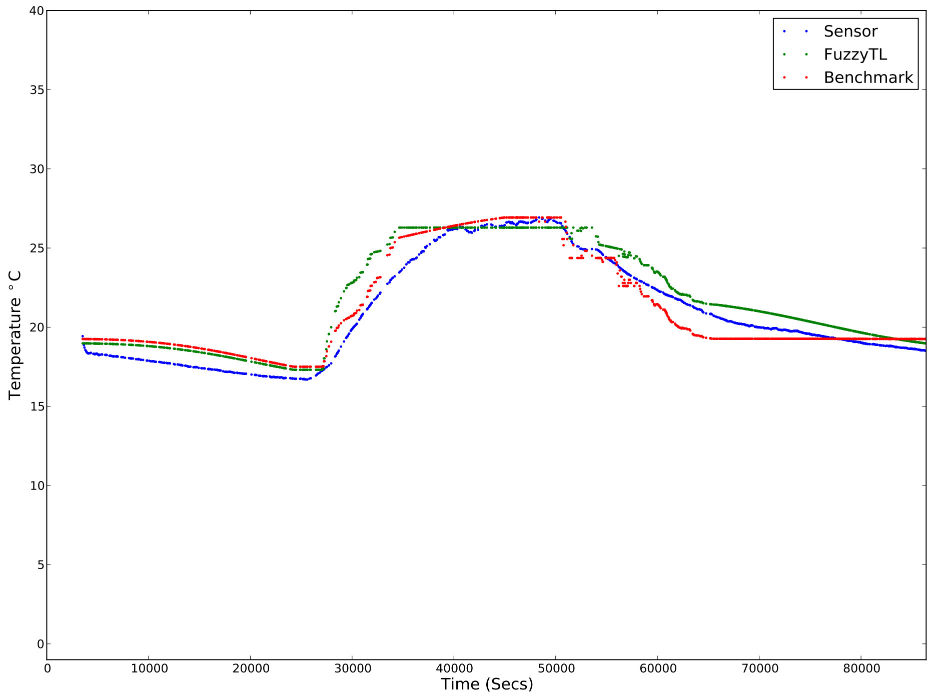

Figure. 4.7. Comparison of Benchmark and Best FuzzyTL to Sensor Readings *Target Data Sensor 34, 28th February, 2004*.

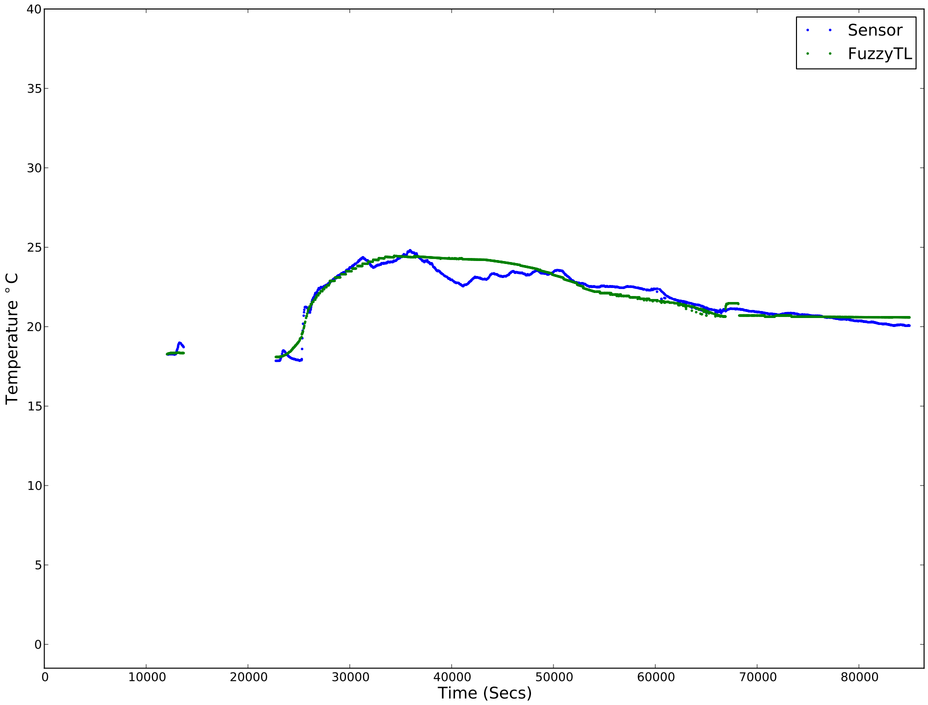

Figure. 4.8. Comparison of FuzzyTL to Sensor Readings *Source Data Sensor 42, 3rd March, 2004 and Target Data Sensor 7, 2nd March, 2004.*

framework to be flexible.

Of the 2352 contexts, 66.1990% (1557 out of 2352) produced a RMSE that was equal, or within the minimum and maximum interval of the benchmark dataset. This indicates that the FuzzyTL was able to use differing contextual source data to produce comparable predictive output. The lowest of those contexts produced a RMSE of $0.5139°$C. The data used was source data from sensor 42 on the 3rd March, 2004 and target data from the sensor 7 on the 2nd March, 2004. Figure 4.8 shows the performance of the FuzzyTL against the actual sensor reading. Figure 4.8 highlights that the FuzzyTL framework output is consistently close to the recorded value of the sensor.

In comparison, the highest error produced was a RMSE of $10.5014°$C. The source data was provided by sensor 7 on the 4th March, 2004, and the target data by sensor 24 on the 2nd March, 2004. Figure 4.10 shows the overall performance. The returned RMSE can be attributed to the nature of the target data. The different structure of the interval domains of the source and target consequent sets, produced a variation in the output compared to the actual value. This is shown in

Figure 4.9. The highest source consequent had a left limit value y_L^{s1} of $17.2640°\text{C}$. The right limit

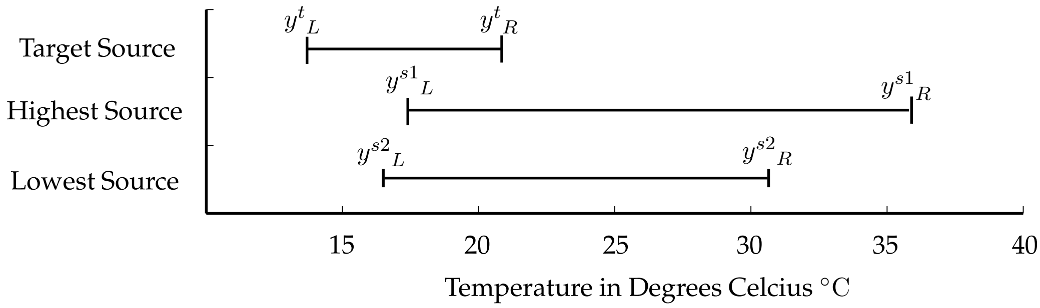

Figure. 4.9. Comparison of the Consequent Intervals For Sensor 24, 4th March, 2004 (Highest Source), Sensor 42 on the 3rd March, 2004 (Lowest Source) and Sensor 24 on the 2nd March, 2004 (Target)

value y_R^{s1} was $36.1584°\text{C}$. The target interval was an intersection of this interval. The left target limit value y_L^t was $14.3044°\text{C}$ and the right target limit was $21.2624°\text{C}$. In contrast, the lowest error rate had a closer consequent domain interval. The source had a left limit y_L^{s2} of $16.9308°\text{C}$ and a right limit y_R^{s2} of $31.5034°\text{C}$. The target had a left limit of $17.8520°\text{C}$ and a right limit of $24.8100°\text{C}$. The knowledge in the consequent domain is dependent on the source task. The closer the source and target consequent domain intervals, the smaller the RMSE that is produced. The impact of the structure of the source and target domain interval is discussed further in Section 4.4.

4.3.2 Robotics Laboratory Data Comparison to Observed Values

A similar analysis was undertaken using the Robotics Laboratory dataset. Benchmark output was created using the adapted WM system. Source data was provided directly from the target domain. The adapted WM methodology produced 12 RMSE values (four days $\times$ three sensors) based on the sensors in the Robotics laboratory and across the defined number of days. These values were compared to the lowest, and so best, output produced by the FuzzyTL framework using different contextual data. Initially the benchmark of the Robotics Laboratory Data and the output of the FuzzyTL were tested for normality. As with the Intel Laboratory dataset Anderson-Darling tests were used. Tables 4.10 and 4.11 show the results of the tests. In both datasets an alpha value α of 0.05 was used. The p-value in both cases was calculated to be higher than the α value, showing the data does not depart from normality.

As with the Intel dataset, a paired t-test was carried out. An α value of 0.05 was used for the test. The results of the paired t-test are shown in Table 4.12. The p-value that was produced was lower than the defined α significance value. This rejects the null hypothesis that the Robotics Laboratory benchmark dataset is the same distribution as the best output of the FuzzyTL

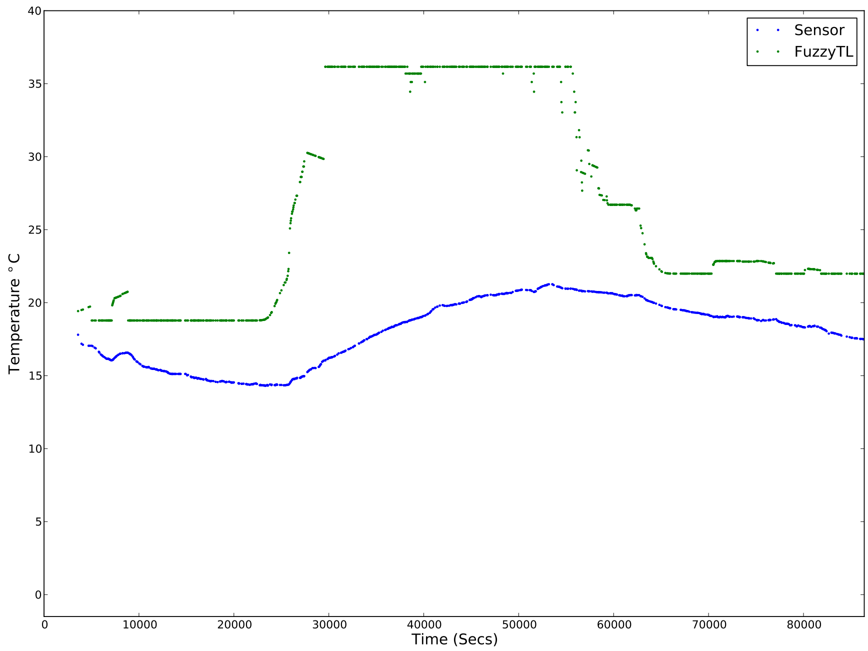

Figure. 4.10. Comparison of FuzzyTL to Sensor Readings *Source Data Sensor 24, 4th March and Target Data Sensor 12, 28th February*.

Anderson-Darling Test	
Data: Benchmark Robotics Laboratory	
Alpha-value	0.05
P-value	0.1519
Observation Number	12
Conclusion	Possibly Normal

Table 4.10. Anderson-Darling Test Results For the Benchmark Robotics Laboratory Output.

Anderson-Darling Test	
Data: Best FuzzyTL Robotics Laboratory	
Alpha-value	0.05
P-value	0.8909
Observation Number	12
Conclusion	Possibly Normal

Table 4.11. Anderson-Darling Test Results For the Best FuzzyTL Robotics Laboratory Output.

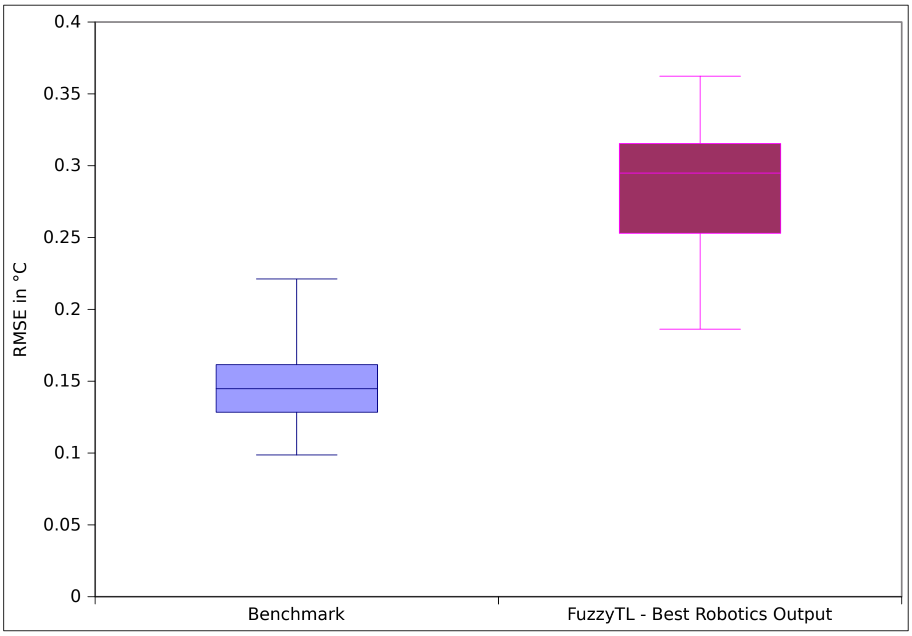

Figure. 4.11. Comparison of Adapted Wang-Mendel Benchmark and Best FuzzyTL Output Using the Robotics Laboratory Dataset.

framework. Further analysis showed that all contexts produced a higher RMSE value than the benchmark (100%).

Paired T-Test	
Data: Robotics Laboratory Benchmark Compared to Robotics Laboratory FuzzyTL	
df	11
T-Statistic	-11.8681
Alpha-value	0.05
P-value	1.3030^{-7}
Positive Differences	12
Negative Differences	0

Table 4.12. Paired T-Test Results For the Robotics Laboratory Benchmark and Best FuzzyTL Output.

Figure 4.11 illustrates the benchmark dataset against the FuzzyTL frameworks best values. This graphically shows the difference between the two datasets. The medians are far apart combined with the FuzzyTL output overall being more spread. The difference between the 1st and 3rd quantiles of the FuzzyTL output is greater than the benchmark, a RMSE of $0.1862°\mathrm{C}$ and $0.0987°\mathrm{C}$, and $0.3623°\mathrm{C}$ and $0.2212°\mathrm{C}$ respectively. The values expressed in Figure 4.11 are shown in Table 4.13.

	Robotics Benchmark	Robotics Best FuzzyTL
Min	0.0987	0.1862
Max	0.2212	0.3623
Median	0.1448	0.2948
Quantile 25%	0.1284	0.2529
Quantile 75%	0.1615	0.3154

Table 4.13. Data From Comparison of Adapted Wang-Mendel Benchmark and Best FuzzyTL Output Using the Robotics Laboratory Dataset.

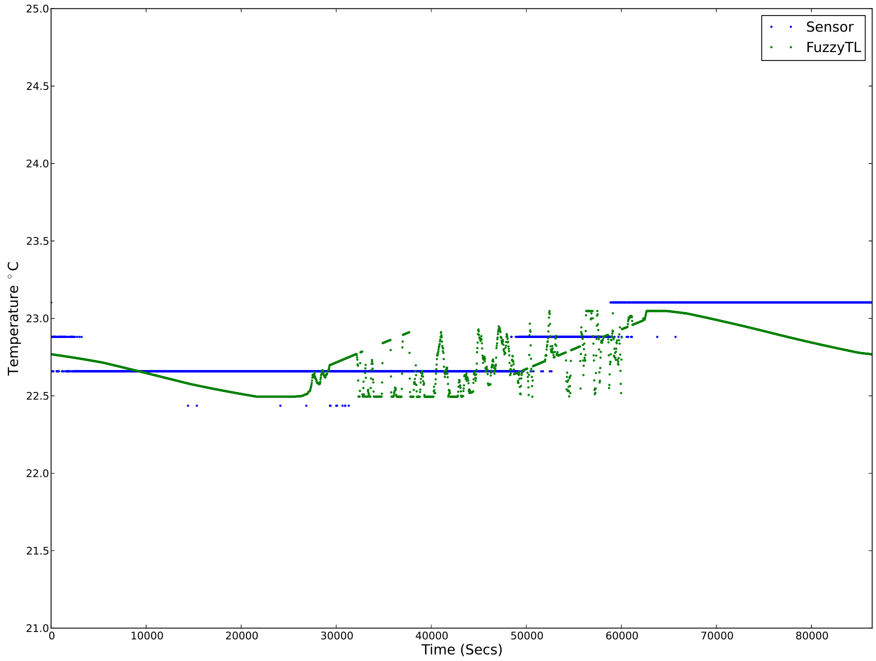

Figure. 4.12. Comparison of FuzzyTL to Sensor Readings *Source Data Sensor 3, 17th October, 2004 and Target Data Sensor 3, 18th October, 2004.*

Focussing closer on the data however, individual source contexts produced RMSE values that were comparable to the benchmark and similar to the actual sensor output. Overall, the lowest RMSE value produced was using source data from sensor 1 on the 18th October 2011, to predict values for sensor 1 on 17th October 2011. The RMSE value for the context was 0.1862. Figure 4.12 depicts the output in detail. The benchmark produced RMSE values in the range of a RMSE of 0.0987 to $0.2212°\mathrm{C}$. The RMSE for this context is $0.0987°\mathrm{C}$. The Robotics laboratory produces a

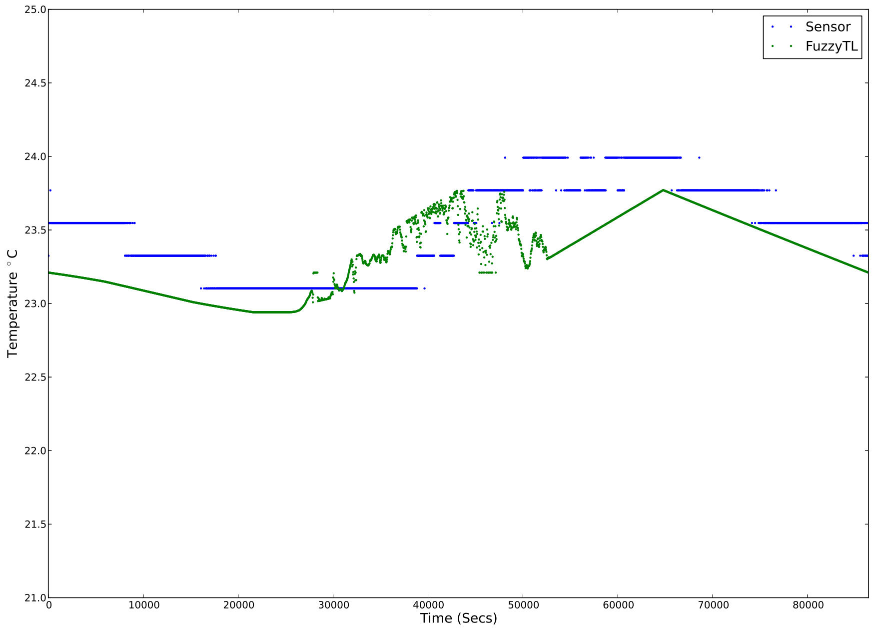

Figure. 4.13. Comparison of FuzzyTL to Sensor Readings *Source Data Sensor 2, 16th October, 2004 and Target Data Sensor 1, 17th October, 2004.*

more consistent temperature than found within readings recorded in the Intel dataset. The variation across the context shown in Figure 4.12 is $0.67°\mathrm{C}$. The FuzzyTL framework replicates the narrow variation producing a value of $0.55°\mathrm{C}$. The initial value calculated, based on zero data, produced an error of only $0.11°\mathrm{C}$. Figure 4.12 illustrates that the FuzzyTL framework is able to again predict the value of a sensor output based on no prior labelled, and little or no unlabelled data.

The highest RMSE produced was a value of 3.3447, using source data from sensor 2 on the 16th October, 2011, to predict values for sensor 1 on 17th October, 2011. Figure 4.13 shows this result in detail. The variation across both the output of the sensor and predictive value is again low, 0.89 and $0.83°\mathrm{C}$ respectively. The impact of the size of the output interval will be discussed in the following section. Despite the RMSE value, Figure 4.13 shows that a similar pattern is formed between outputs.

4.3.3 Summary of Results

A number of conclusions were drawn from the experiments carried out in previous sections. These can be summarised as:

Intel Laboratory Benchmark Comparison

- The best performing Intel FuzzyTL predictive output was significantly different to the Intel Laboratory benchmark.
- Of those differences, 22.4490% were a lower RMSE than the benchmark.
- The FuzzyTL framework was shown to be able to use different contextual information to output comparable values to a system with prior knowledge.

Robotics Laboratory Benchmark Comparison

- A number of different source contexts were able to produce predictive output comparable to the benchmark dataset.
- The best performing Robotics Laboratory data was significantly different to the benchmark dataset that was produced.
- The differences were shown to be higher RMSE values.
- Overall, the FuzzyTL framework did not perform as well using the Robotics Laboratory dataset. This can be possibly attributed to the lower variation in the data.

4.4 Context Impact

The transfer of information is dependent on the context of the data. Through a series of experiments, the impact of the context will be investigated. Two context sub-types will be used: inter and intra.

4.4.1 Inter Contextual Experiments

As defined previously, inter contexts are based on a single scenario. For the purposes of the experiments in this section, two separate intra contexts will be used, the Intel laboratory, and the Robotics laboratory. For the inter contexts a contextual distance metric was calculated. Using this metric, the impact of the contextual distance on the ability of the FuzzyTL framework to predict values was assessed.

Evaluation of Contexts Across the Intel Laboratory Dataset To understand the way CD impacts the FuzzyTL framework, the difference in distance of the source and target contexts of each scenario was examined. Initially, an assessment was carried out focussed on the highest CD results. Of the 2352 contexts, the greatest distance was 4.8603. Two contexts were measured at

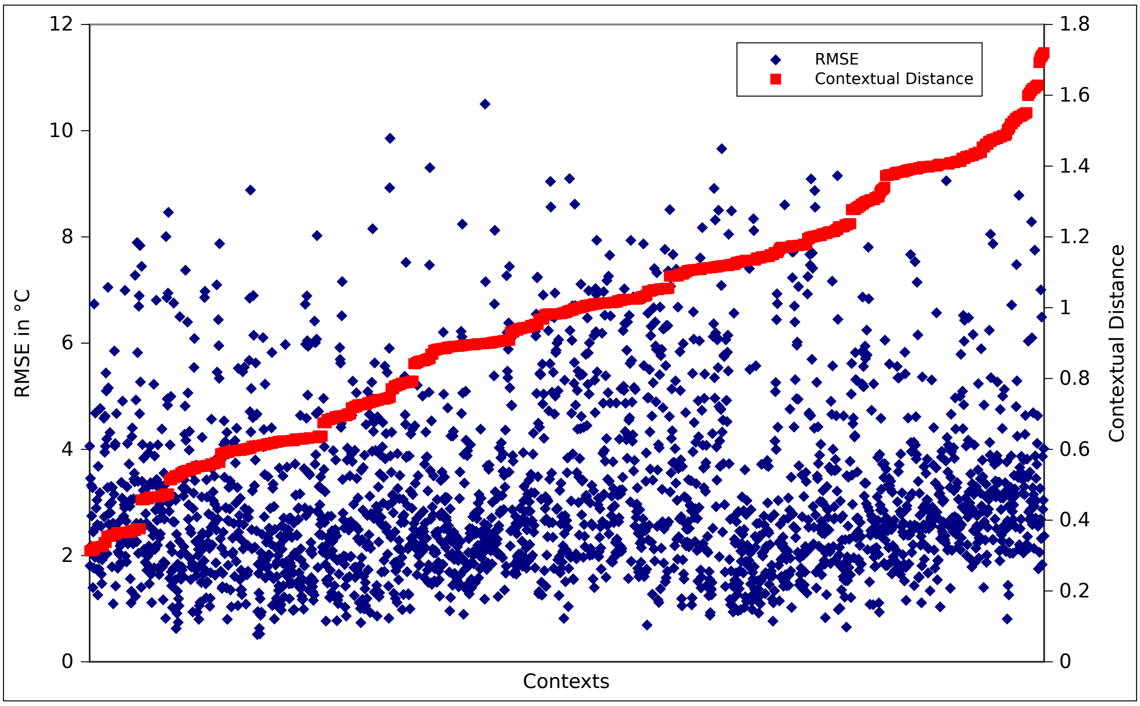

Figure. 4.14. Comparison of RMSE to Context Distance For Intel Laboratory Dataset.

this distance. The first context (sensor 12 on the 28th February, 2004 used to predict values of target data sensor 42 on 5th March, 2004) produced an RMSE of 4.0123°C. The highest RMSE produced for the target sensor 42 on 5th March, 2004 is 6.5694°C. The CD for this value was 4.3130.

The second (sensor 12 on 3rd March, 2004 used to predict values of target data sensor 42 on 28th February, 2004) produced an RMSE of $3.4951°\mathrm{C}$. The highest RMSE for this target was an RMSE of $9.0564°\mathrm{C}$ based on a CD of 4.1465. These contexts illustrate that higher CD does not produce the greater error. The system absorbs the changes in the contexts. The lowest CD values indicate similar findings. A Pearson correlation of the CD and the RMSE (resulting in an output of 0.0775) shows that there appears to be very little link between the two values. Figure 4.14 shows the comparison of RMSE values against the CD. As the CD increases, the RMSE continues to remain within a similar distribution. The highest proportion of the contexts have an RMSE value of 0.5 or below despite the increasing contextual distance. From this analysis, an initial conclusion can be drawn. CD plays a small part in the output of the FuzzyTL framework. Inter contextual changes are absorbed within the system through the use of both a fuzzy and adaptive methodology.

CD contains each of the input interval domains. The differences that occur in the input interval domains are absorbed by the FuzzyTL framework. Figure 4.15 shows the difference between the

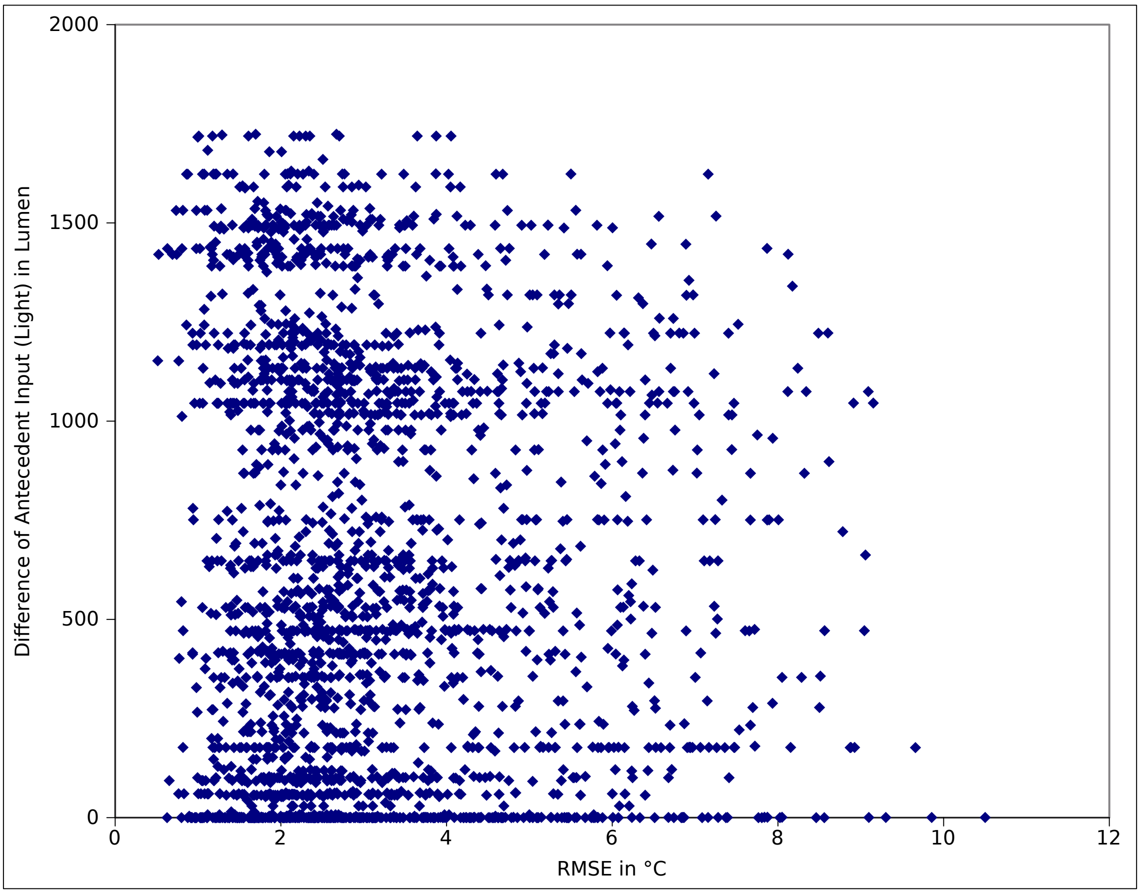

Figure. 4.15. Difference of Source and Target Light Input Against the RMSE Output of the Intel Laboratory Dataset.

source and target light values, and the corresponding RMSE. The difference in values is calculated by taking the absolute difference of source minimum s_{min} and target minimum t_{min}. A Pearson Correlation of the light input difference and the RMSE output produces a value of -0.02904. This highlights that there is no correlation between the light input and the RMSE. The input values are absorbed by the adaptation process of the FuzzyTL framework.

There is a stronger correlation between the consequent (output) values and the RMSE produced. Figure 4.16 shows the difference between the source and target temperature values, and the corresponding RMSE. A Pearson correlation of the same datasets produces a value of 0.6735.

Differences between the source and target output interval domain can have a impact. Larger initial differences in the consequent (output) domain have a larger impact. The unknown nature of the output interval domain can produce negative learning. The FuzzyTL framework uses a

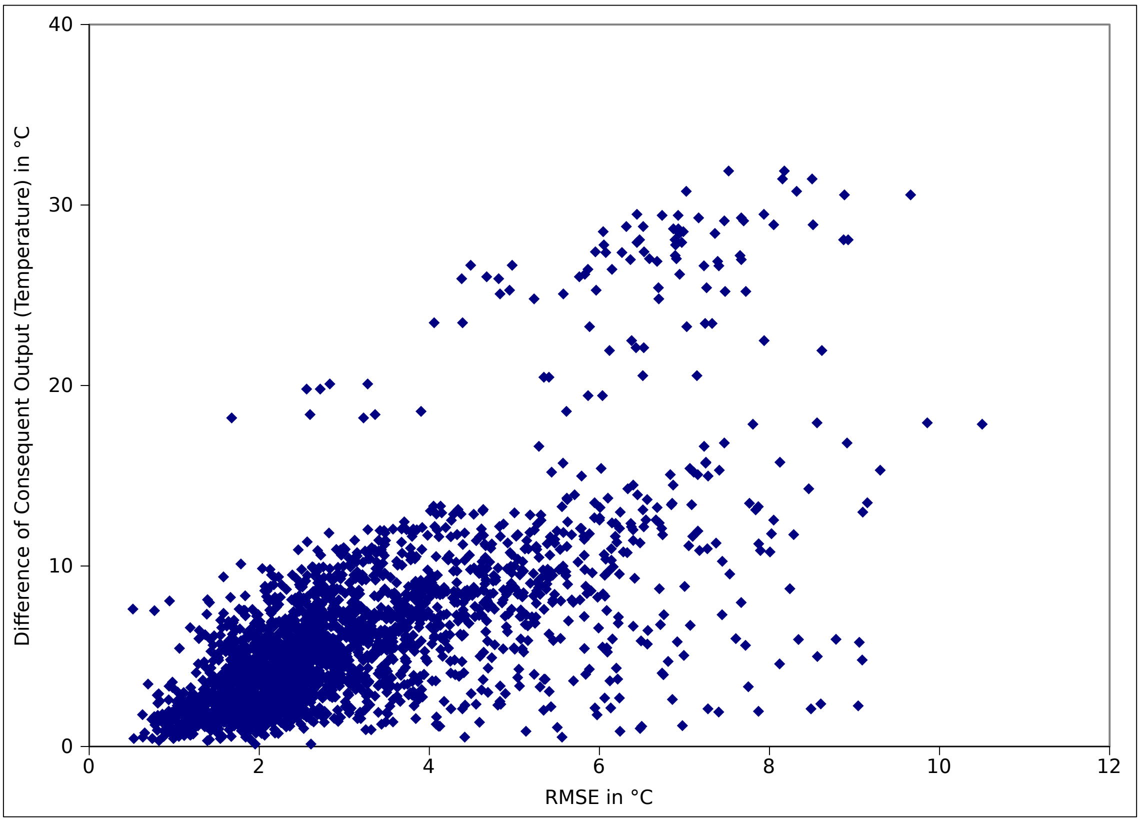

Figure. 4.16. Difference of Source and Target Temperature Output Against the RMSE Output of the Intel Laboratory Dataset.

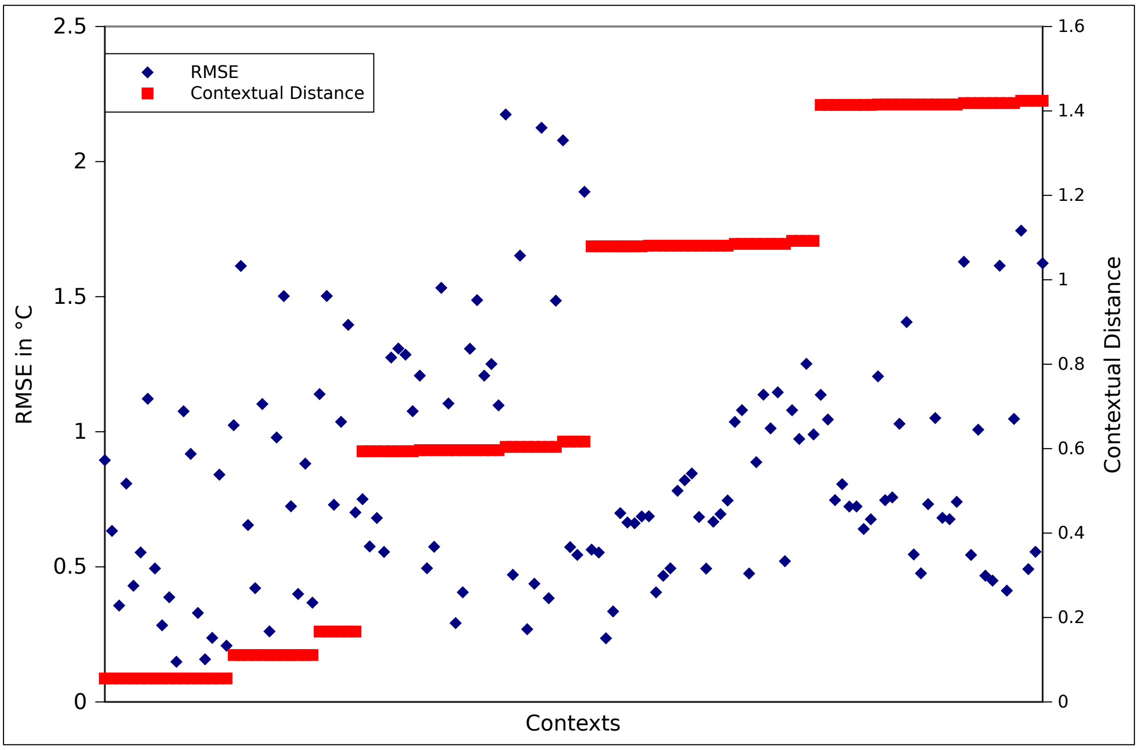

Figure. 4.17. Comparison of RMSE to Context Distance For Robotics Laboratory Dataset.

feedback system to understand the adaptations made to the consequent (output) domain. This is led by the source data. If the correlation between the source and target consequent data is weak, it can lead to misdirected learning. The FuzzyTL may interpret the combination of source and target data incorrectly but be unaware of any issues. This is a short coming of the system, but is bound by the nature of the data structure.

Aggressive adaptive strategies can exasperate this situation. A strategic decision is necessary to assess which approach is appropriate. More aggressive strategies can result in greater performance, but equally higher negative impact can occur. To reduce negative impact, a more lenient strategy is employed within the FuzzyTL framework.

Evaluation of Contexts Across Robotics Laboratory Dataset The smaller dataset from the Robotics laboratory produced similar results to the Intel laboratory data. Figure 4.17 shows a diagram of a comparison of contextual distance to RMSE output. Less dense than the Intel distribution, the Robotics laboratory comparison still highlights that an increasing distance does not relate to an increase in RMSE. The adaptive nature of the FuzzyTL is able to compensate for the difference in contexts that manifests itself within the input and output domain intervals.

4.4.2 Intra Contextual Experiments

To further understand the impact of contexts on the FuzzyTL framework, an intra-contextual experiment was carried out. Using the Robotics laboratory dataset as source, the framework was used to predict the temperature values of the Intel laboratory WSN. The same structure was isolated as used in the previous experiments. 588 separate contexts were used for comparison against the known sensor readings. Based on the intra structure as outlined in Section 4.2.3, no contextual distance was produced for this comparison.

To assess the overall performance of the Robotics laboratory dataset in providing a starting learning structure, the best RMSE values were taken from the intra context system (composed of source data from the Robotics laboratory) and compared this to the best values of the inter context system (composed of Intel laboratory source data). This produced 49 separate contexts. The two datasets were initially tested for normality using Anderson-Darling tests. It was found that the intra dataset was non-normal. To normalise the data a power transform was used. This took each value x and applied $log_1 0$. The results for both sets of data are shown in Tables 4.14 and 4.15. Both results showed that the calculated p-value was higher than the defined alpha α value showing the datasets to be possibly normal.

Anderson-Darling Test	
Data: Benchmark Robotics Laboratory	
Alpha-value	0.05
P-value	0.8663
Observation Number	49
Conclusion	Possibly Normal

Table 4.14. Anderson-Darling Test Results For the Best Inter Intra Laboratory Output.

Anderson-Darling Test	
Data: Best FuzzyTL Robotics Laboratory	
Alpha-value	0.05
P-value	0.5644
Observation Number	49
Conclusion	Possibly Normal

Table 4.15. Anderson-Darling Test Results For the Best Inter Intel Laboratory Output.

Following the normality tests, a paired t-test was run across the data. The paired t-test showed that the datasets came from different distributions. The results are shown in Table 4.16. Analysing the differences, these showed that none of the intra contexts were lower than the inter data contexts. It can be inferred that the intra source data did not produce an output that was as comparable in

Paired T-Test	
Data: Best Intra Intel Laboratory Compared to Best Inter Intel Laboratory Output	
df	48
T-Statistic	22.0644
Alpha-value	0.05
P-value	4.4670^{-27}
Positive Differences	49
Negative Differences	0

Table 4.16. Paired T-Test Results For the Best Intra Intel Laboratory Compared to Best Inter Intel Laboratory Output.

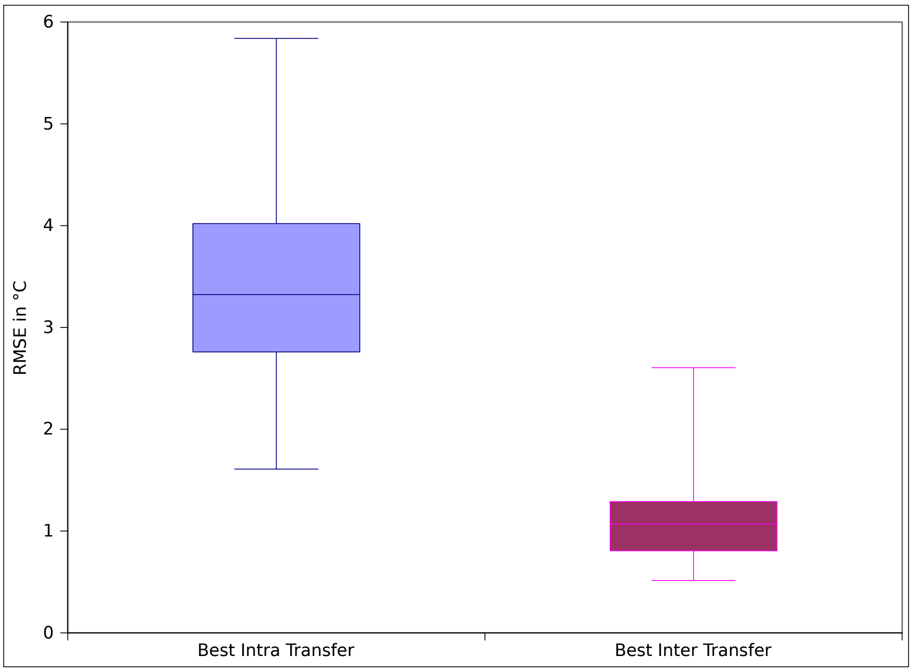

Figure. 4.18. Comparison of Best Inter RMSE Output to the Best Intra RMSE Output.

performance to the inter lowest inter source values. The paired t-test highlighted that the difference of the means was $2.2737°\mathrm{C}$ (RMSE values of 1.1285 and $3.4023°\mathrm{C}$ respectively). The variance of the intra source was considerably higher than the inter source, an RMSE of $0.9857°\mathrm{C}$ compared to $0.1823°\mathrm{C}$. This can be seen in Figure 4.18. Figure 4.18 shows that the lower and upper quartile are more compressed and focussed around a RMSE of $1.0°\mathrm{C}$. The intra output is move spread and focussed around a RMSE of $3.0°\mathrm{C}$. This analysis shows that the intra source does not perform as well as the inter source process.

Further analysis of the dataset illustrates that despite the differences in context, elements of the intra source information allows predictive output to be produced that is comparable to the

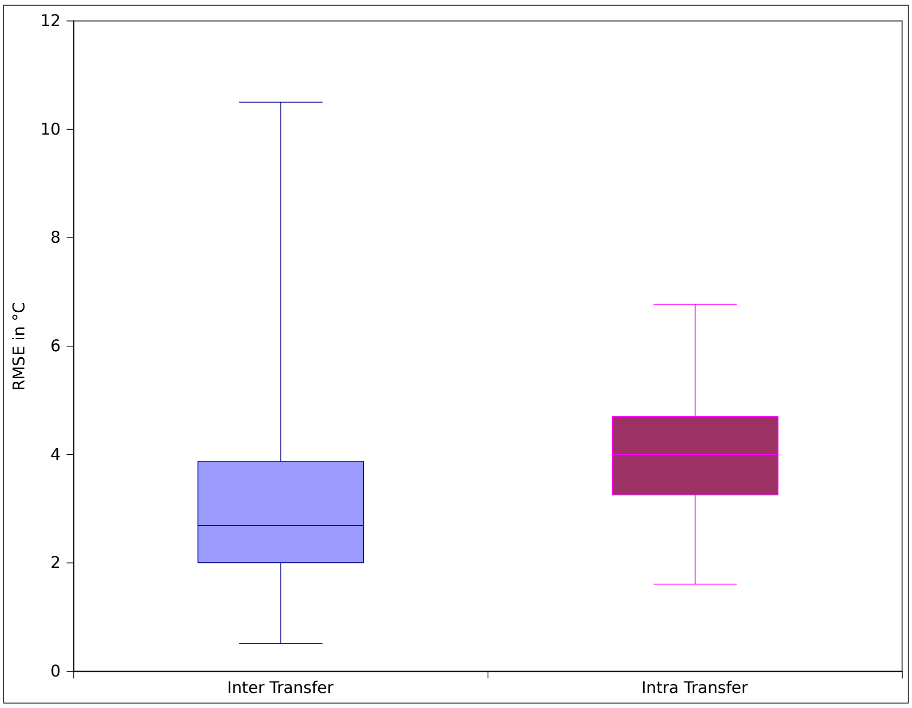

Figure. 4.19. Comparison of Intel to Intel RMSE Output to the Robotics to Intel RMSE Output.

inter source process. A comparison of the whole datasets of both the inter and intra shows that the variation of the inter source process is higher than the intra process (RMSE of $2.7669°\mathrm{C}$ compared to a RMSE of $1.1119°\mathrm{C}$). Figure 4.19 shows a comparison of the two datasets. The figure highlights that the maximum value of the intra process is lower than the inter process. The lower and upper quartile are shown to be less spread than the inter process although the median is greater.

Examining the data further, the lowest calculated RMSE value for the intra dataset resulted from the source data of sensor 2 on 17th October, 2011, used to predict values of the target data sensor 7 on 2nd March, 2004. The RMSE produced was $1.6085°\mathrm{C}$. In comparison, the inter source process produced values that ranged from a RMSE of $0.5139°\mathrm{C}$ to $7.1579°\mathrm{C}$. The median of the values produced for the target was a RMSE of $2.097°\mathrm{C}$. Figure 4.20 shows the output of the lowest intra process against the actual sensor output.

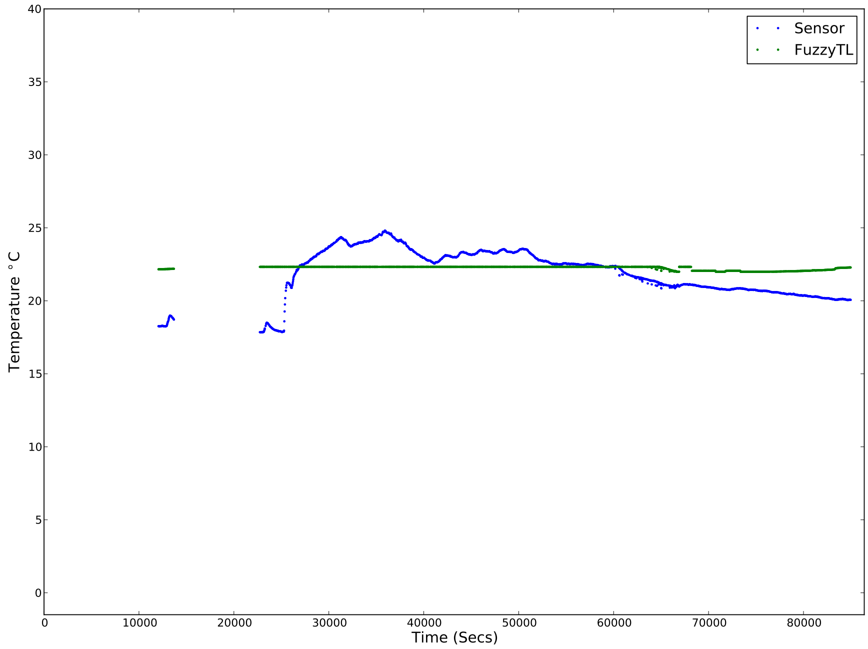

Figure. 4.20. Comparison of FuzzyTL to Sensor Readings *Source Data Sensor 2, 17th October, 2011 and Target Data Sensor 7, 2nd March, 2004.*

The FuzzyTL framework demonstrated that although it is unable to match the performance of inter contexts, the output of intra contexts is partially comparable despite the distinct contextual variation.

4.4.3 Summary of Results

Through looking at the contextual nature of the data, key findings were made. These were:

Inter Contextual Comparison

- Increases in CD do not have the same impact on the RMSE output of the FuzzyTL framework.
- The differences that occur in the input interval domains can be absorbed by the FuzzyTL framework.
- Greater differences in the consequent values of the source and target data can produce higher RMSE values.

Intra Contextual Comparison

- Source data from intra contexts can be used to produce predictive output based on unlabelled target information.
- The performance of the intra contexts in predicting output is reduced compared to inter contexts.

4.5 Adaptation

In this section, a series of experiments are shown that were constructed to test the second posed hypothesis. The FuzzyTL framework is grounded on the transfer and adaptation of information. To investigate the performance gain through the use of the adaptation process, a comparison was made between a non-adapted system, and the full FuzzyTL framework.

4.5.1 Comparison of Non-Adaptive FuzzyTL to Full FuzzyTL Framework: Intel Laboratory Data

The non-adaptive system was composed of a transferred Fuzzy Inference System (FIS). The learning processes involved in forming the structure of the fuzzy system remained unchanged to those previously used. Supplementary online adaptation and learning was removed providing a base to compare to. The first experiment was based on the Intel laboratory dataset. The structure outlined in Section 4.2.1 was used to form the basis for the comparison. Using a similar performance metric, each of the different contexts were compared to the sensor readings. The non-adaptive RMSE values were subsequently compared to the values previously gained from using the full FuzzyTL system. A total of 2352 contexts were used for comparison.

To compare the two sets of data, both were initially assessed for normality. Previously a sample of the Intel Laboratory dataset was taken matching a distinct criteria. This experiment used the whole dataset. Equally, the whole of the non-adapted dataset was processed. Tables 4.17 and 4.18. For both datasets an alpha value α of 0.05 was used. This indicates the predetermined significance level of the test. The p-value for both datasets was lower than the α value indicating the null hypothesis is rejected.

Through the application of a number of power transforms, normality of both datasets could not be achieved. To compare the two datasets, a Wilcoxon signed-rank test was chosen. The Wilcoxon signed-rank test provides a non-parametric alternative to the paired t-test when the sample populations are non-normal. The test is applicable to the adapted and non-adapted Intel Laboratory datasets. The results are shown in Table 4.19. The p-value for the Wilcoxon signed-rank test is lower than the defined alpha value. This indicates that the two datasets are

Anderson-Darling Test	
Data: Adapted Intel Laboratory	
Alpha-value	0.05
P-value	2.9371^{-141}
Observation Number	2352
Conclusion	Not Normal

Table 4.17. Anderson-Darling Test Results For the Adapted Intel Laboratory Output.

Anderson-Darling Test	
Data: Non-Adapted Intel Laboratory	
Alpha-value	0.05
P-value	1.3390^{-112}
Observation Number	2352
Conclusion	Not Normal

Table 4.18. Anderson-Darling Test Results For the Non-Adapted Intel Laboratory Output.

Wilcoxon Signed Rank Test	
Data: Intel Laboratory Non-Adapted Compared to Intel Laboratory Adapted	
Alpha-value	0.05
V-Value	132366.5
P-value	0
Observation Number	2352

Table 4.19. Wilcoxon Signed Rank Test For Adapted and Non-Adapted Intel Laboratory Output.

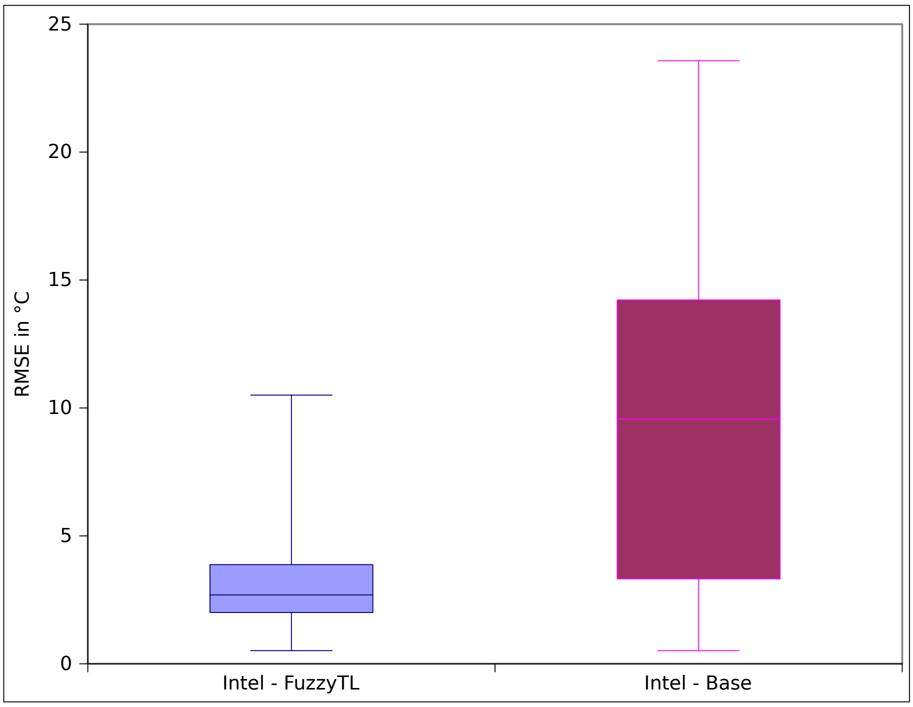

Figure. 4.21. Comparison of Base And Adapted FuzzyTL Framework Using the Intel Dataset.

from different distributions. Further analysis showed that 87.6700% (2062 of 2352) of contexts exhibited a decrease in RMSE when the adaptation was applied. These differences can be seen in Figure 4.21. The highest decrease was an RMSE of $20.9288°\mathrm{C}$. Of the 12.3299% of contexts that showed an increase in RMSE, the greatest was $6.0450°\mathrm{C}$. This analysis illustrates that the use of the adaptation stages within the FuzzyTL decreases the error rate produced.

The difference between the two datasets can be seen in Figure 4.21. The difference between the 1st and 3rd quartile is far greater for the base dataset than the adapted FuzzyTL. The median of the base dataset was an RMSE of $9.5721°\mathrm{C}$, a difference of $6.8797°\mathrm{C}$ to the median of the FuzzyTL framework.

Isolating a single context comparison, Figure 4.22 shows the comparison of the non-adaptive and full FuzzyTL systems using source data from sensor 7 on 2nd March, 2004, and target data from sensor 24 on 2nd March, 2004. This illustrates the non-adaptive systems inability to cope with the initial prediction, producing a -1 value. This is due to the nature of the input, and output domain intervals. A comparison of the input and output interval domains can be seen in Table 4.20.

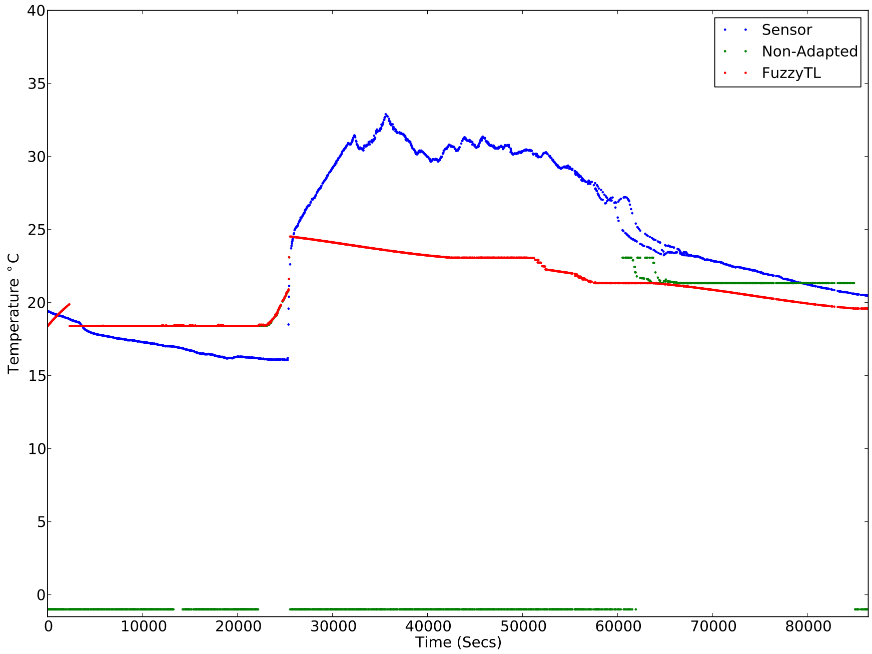

Figure. 4.22. Comparison of Non-Adapted System, FuzzyTL and Sensor *Source Data Sensor 7, 2nd March, 2004 and Target Data Sensor 24, 2nd March, 2004.*

	t_L	t_R	l_L	l_R	tm_L	tm_L
Target Sensor	51	86396	97.52	426.88	17.77	20.92
Non-Adapted	7	86107	158.23	1788.49	17.84	30.78
FuzzyTL	51	86396	97.52	426.88	17.81	30.75

Table 4.20. Comparison of Input and Output Interval Domains

Table 4.20 describes the input interval domains where t is time, l is light and tm is temperature. L and R are the left and right limit of the interval. Issues arise within the non-adaptive system as the input values fall outside of the interval domain. The minimum of the target sensor interval (l_L^{ts}) sits outside of the non-adapted light interval (l_L^{na}), $l_L^{na} > l_L^{ts}$.

The results of this are shown in Figure 4.22 between 51 to 27260 seconds (00:00:51 - 07:34:20). The non-adaptive system can not produce an output based on these inputs. As the inputs move into the domain of the non-adapted system, an output is produced. This can be seen from 27260 seconds onwards. At this point the FuzzyTL framework adapts the input domains based on the new data. This allows it to output a value. This occurs again at 61136 to 61470 seconds and 61610 to 86396 seconds.

Taking on board the transferred information, and the new data, the FuzzyTL framework adapts the output interval domain. Driven by the inputs from both the target and source data, the output interval domain is shown, in this example, to move toward the parameters of the sensor interval. Overall, the incorporation of the adaptation stages into the FuzzyTL improved the predictive capability when used with the Intel Laboratory dataset.

The structure of the target domain data can produce positive results when using only the non-adapted FuzzyTL. The non-adaptive system is shown to perform well when the target input and output interval domains are proper subsets of the source. This is illustrated by Table 4.21. The

	t_L	t_R	l_L	l_R	tm_L	tm_L
Non-Adapted	85	86398	0.02	1376.32	16.93	31.50
Target Sensor	12059	84959	1.38	224.48	17.85	24.81

Table 4.21. Comparison of Input and Output Interval Domains for Source Data Sensor 42, 3rd March, 2004 and Target Data Sensor 7, 2nd March, 2004.

table shows the domains of the lowest context that produced the lowest RMSE output, sensor 42 on 3rd March, 2004 used as the source to predict values from sensor 7 on the 2nd March, 2004. An RMSE of $0.5144°\mathrm{C}$ was calculated for this context using the non-adapted system. The input domain intervals of the non-adaptive system show that they are contain the target sensor values. This can be defined as $T^{na} \subseteq T^{ts}$, $L^{na} \subseteq L^{ts}$ and $TM^{na} \subseteq TM^{ts}$ were na is the non-adaptive system, ts is the target sensor, and T, L and TM are the time, light and temperature intervals respectively. The containment of the target values within the source intervals reduces error. The dynamic nature of the data is absorbed by the underlying fuzzy inference structure, removing any need for adaptation. Figure 4.23 depicts the output of the non-adapted system contrasted with the actual sensor reading.

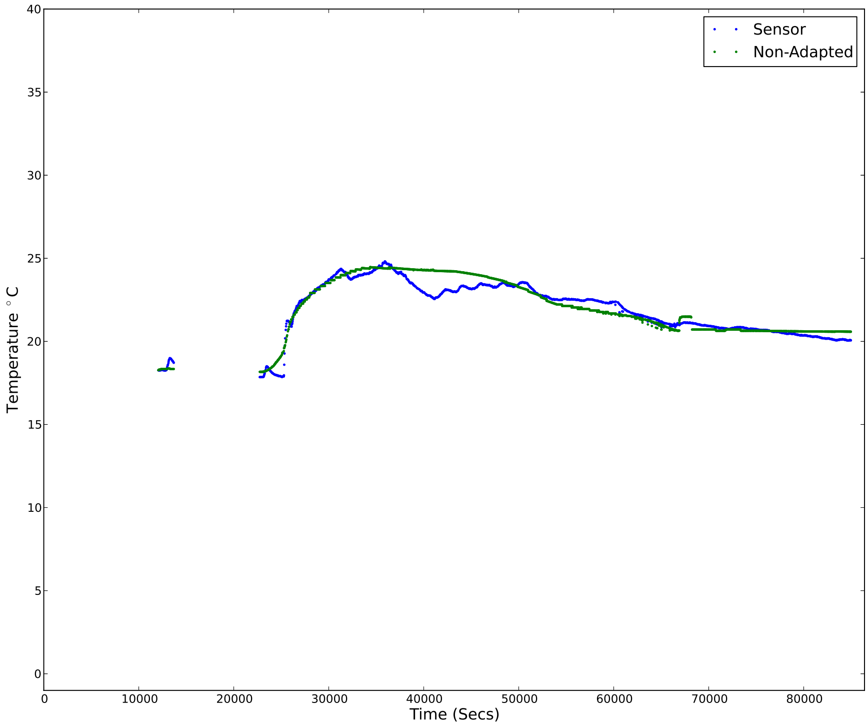

Figure. 4.23. Comparison of Non-Adapted System, FuzzyTL and Sensor *Source Data Sensor 42, 3rd March, 2004 and Target Data Sensor 7, 2nd March, 2004*.

4.5.2 Comparison of Non-Adaptive FuzzyTL to Full FuzzyTL Framework: Robotics Laboratory Data

In a similar approach to the Intel laboratory dataset, a comparison was made between a non-adaptive and full FuzzyTL system using the Robotics laboratory data. The non-adaptive system was based on the FuzzyTL framework with the removal of the adaptive stages.

Both sets of data were firstly assessed for normality using an Anderson-Darling test. Tables 4.17 and 4.18 show the results. For each dataset an alpha value α of 0.05 was used. The p-value for both datasets was lower than the α value indicating the data is not from a normal distribution.

In a similar process to the Intel Laboratory dataset, a number of power transforms were applied to the data. Despite this, both datasets were shown to be non-normal. To compare the data, a Wilcoxon signed-rank test was used. The results are shown in Table 4.24. The p-value calculated was lower than the alpha value defined. Taking the null hypothesis stated that both datasets come

Anderson-Darling Test	
Data: Adapted Intel Laboratory	
Alpha-value	0.05
P-value	2.9371^{-141}
Observation Number	2352
Conclusion	Not Normal

Table 4.22. Anderson-Darling Test Results For the Adapted Intel Laboratory Output.

Anderson-Darling Test	
Data: Non-Adapted Intel Laboratory	
Alpha-value	0.05
P-value	1.3390^{-112}
Observation Number	2352
Conclusion	Not Normal

Table 4.23. Anderson-Darling Test Results For the Non-Adapted Intel Laboratory Output.

from the same distribution, this can be rejected. The differences between the datasets showed

Wilcoxon Signed Rank Test	
Data: Intel Laboratory Non-Adapted Compared to Intel Laboratory Adapted	
Alpha-value	0.05
V-Value	1155
P-value	2.0700^{-13}
Observation Number	132

Table 4.24. Wilcoxon Signed Rank Test For Adapted and Non-Adapted Intel Laboratory Output.

that 82.5757% (109 of 132) of the contexts produced a lower RMSE when the adaptive stages were applied. Of those, the greatest reduction was an RMSE of $14.2349°C$. By comparison, of the 17.4242% that produced a higher RMSE, the greatest increase was $1.1318°C$. These findings substantiate the previous conclusions drawn from the Intel Laboratory dataset.

Drilling down into the contexts, Figure 4.24 shows an example of non-adapted system output compared to the actual sensor reading. This figure shows how, in certain conditions, the non-adapted system is unable to output a value for a portion of the target (Within Figure 4.24, the missing output is not displayed).

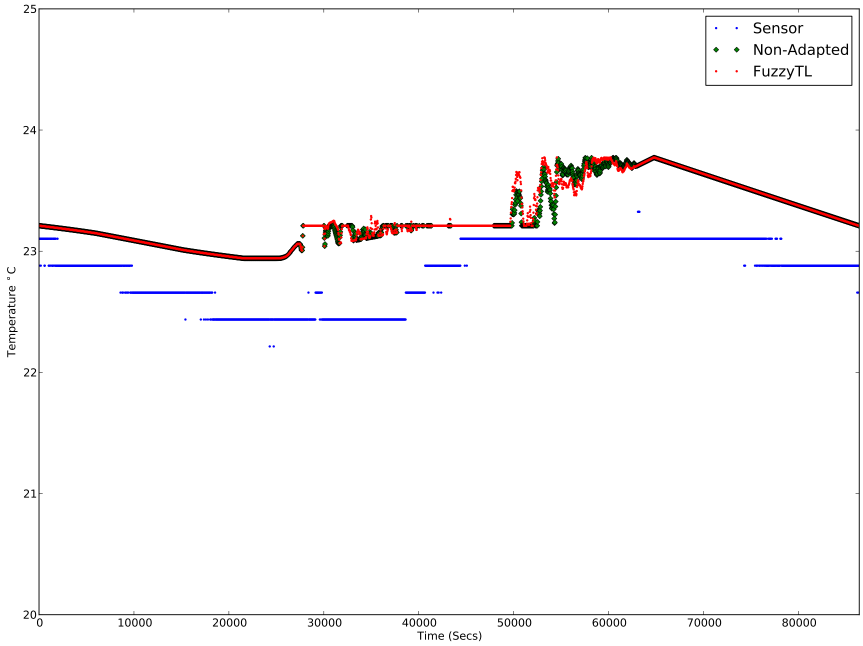

Figure. 4.24. Comparison of Non-Adapted System, FuzzyTL and Sensor *Source Data Sensor 2, 16th October, 2011 and Target Data Sensor 3, 17th October, 2004.*

As the light input value steps outside of the domain, the system fails to produce an output. The transferred light domain is $l_L = -0.01$ and $l_R = 811.01$. At 36842 seconds (10:14:02) the input light level reaches 863, beyond the light domain interval. This causes the system to fail to output a value. Figure 4.24 highlights the points which this occurs.

As with the Intel Laboratory dataset, there are contextual situations where the non-adapted system is able to produce a strong output. Figure 4.25 shows the lowest RMSE output from the non-adapted system. This is using the source data of sensor 3, 18th October, 2011 to predict the values of the target for sensor 3, 17th October, 2011. The RMSE produced for this context was $0.1578°\text{C}$. In comparison, the lowest adapted FuzzyTL for the example target was $0.2920°\text{C}$ produced by the same source and target combination. The adaptive nature of the FuzzyTL framework can have adverse effects on the learning process. These are in the minority. Of the 132 contexts compared, 82.5757% produced a greater RMSE value when non-adapted.

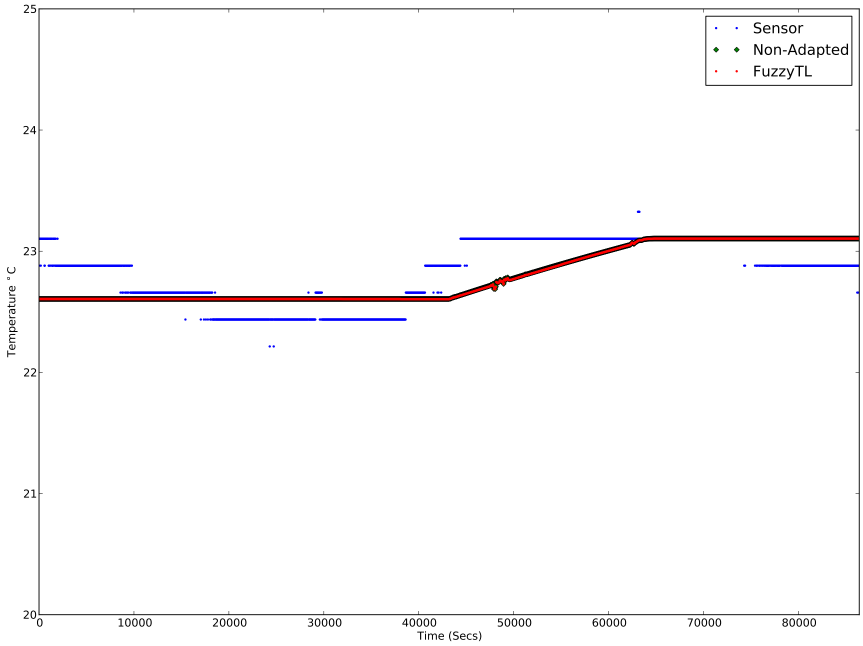

Figure. 4.25. Comparison of Non-Adapted System, FuzzyTL and Sensor *Source Data Sensor 3, 18th October, 2011 and Target Data Sensor 3, 17th October, 2011*.

4.5.3 Summary of Results

From the comparison of the adaptive and non-adaptive processes, a number of conclusions were drawn.

Comparison of Adaptive and Non-Adaptive FuzzyTL framework Using Intel Laboratory Data

- The use of the adaptation stages within the FuzzyTL framework decreases the RMSE produced when applying differing source contexts to target data.
- The non-adaptive process fails to produce output when the target data moves outside of the source domain intervals.
- Close proximity of source and target data allowed the non-adapted process to produce a comparatively low RMSE output.

Comparison of Adaptive and Non-Adaptive FuzzyTL framework Using Robotics Laboratory Data

- The application of the adaptation stages within the FuzzyTL framework on the Robotic Laboratory dataset also decreased the RMSE produced.
- These findings substantiated the previous conclusions drawn from the Intel Laboratory dataset.

4.6 Summary of the Application of Fuzzy Transfer Learning in Intelligent Environments

In this chapter, a number of experiments were set out to test two hypotheses that form a major element of this thesis. To test the hypotheses, the Fuzzy Transfer Learning (FuzzyTL) framework (as defined in Chapter 3) was implemented within a Intelligent Environment (IE) domain. Through the use of multiple Intelligent Environments (IEs), differing contextual situations were demonstrated incorporating a dynamic and uncertain real-world application. The experimental structure was separated into three sections: 1) *Performance*, 2) *Context Impact*, and 3) *Adaptation*. Sections 1-3 formed the basis for the testing of hypothesis 1, with Section 1 being the prime focus. Section 3 tested the second hypothesis.

The findings of this chapter can be summarised in the following three points:

- The FuzzyTL framework can use contextually different but related data to produce predictive output for target tasks.
- Contextual distance has little effect on the Root Mean Squared Error (RMSE) error that is produced by the framework. There is a strong correlation between the size of difference of consequent domain interval distance, and the error produced.
- The adaptation steps of the FuzzyTL framework reduce the RMSE produced when the predictive output is compared to actual sensor readings.

Each experimental step described in this chapter is summarised in the subsequent sections.

4.6.1 Summary of Performance Experimental Process

To test the first hypothesis, a number of experiments were used. Firstly, predictive values were gained from the FuzzyTL framework for each of the two datasets defined in Sections 4.2.1 and 4.2.2. To evaluate the performance of the output, the actual values from the sensor networks were compared to the predictive values. A context was defined for each of the sensors that constituted the part of the network studied, and for the day defined. A RMSE was calculated for each of the contexts. To understand the significance of the RMSE value, the best result gained for each context was evaluated. This subset of the data was compared to a benchmark dataset. The benchmark was produced using the FuzzyTL framework. The source data for these contexts were taken from the same domain as the target task. A paired t-test was carried out to compare the datasets.

Focussing on the Intel Laboratory dataset, overall the performance of the contextually different source FuzzyTL framework was found to be comparable to that of the target source FuzzyTL process. The paired t-test showed that the best values from the source domain FuzzyTL and target domain FuzzyTL were from statistically different distributions. The differences highlighted the FuzzyTL framework was able to output a RMSE value lower compared to the benchmark in 24.4898% of the contexts. This indicated that the FuzzyTL framework could use contextual different information to predict output values to a comparable standard of a system with prior knowledge.

Similar analysis was conducted on the Robotics dataset. This produced results that concurred with the findings of the Intel Laboratory dataset. The exploration of the output produced by the FuzzyTL framework showed it was able to output predictive values that were comparable to the target source process. Overall, the results showed that with zero, or limited unlabelled target data and contextually different labelled source data, the FuzzyTL framework was able to predict sensor values. A comparable accuracy was achieved to a system with knowledge from the same contextual domain.

4.6.2 Summary of Context Impact Experimental Process

Building on the experiments provided in Section 4.3, the impact of different contexts both within, and across the IE datasets were assessed. Using the context definition given in Chapter 2, a contextual distance measure was defined. Based on a normalised euclidean distance, this provided a metric to assess contextual distribution. Used for *inter* contextual differences, a comparison was made between the performance of the FuzzyTL framework, and the contextual distance. The analysis of the Intel Laboratory dataset showed that changes in the inter contextual distance had little effect on the performance of the FuzzyTL framework. Increases in the Context Distance (CD) did not produce a similar increase in RMSE for the predictive value. Differences in the consequent (output) domains produced a greater change within the RMSE output. This can be attributed to the larger knowledge gap that exists when learning unlabelled data. Further comparison, using the Robotics laboratory dataset substantiated this belief.

The FuzzyTL was also applied to *intra* contextually different scenarios. The Robotics laboratory dataset acted as the source, whilst the Intel laboratory provided the target information. A paired t-test was carried out on the lowest output from the FuzzyTL framework using inter and intra source values. This showed that the framework, when using this metric, did not produce predictive output that was comparable to inter sources. However, individual contexts within the intra dataset produced comparable output.

The FuzzyTL framework demonstrated its ability to absorb contextually different information, allowing its use as a catalyst for learning a target task. High context distance and domain interval difference were shown to be absorbed by the adaptation process.

4.6.3 Summary of Adaptation Experimental Process

The second hypothesis was tested using the experiments presented in the third section. Here, a comparison was made between the FuzzyTL and a non-adaptive FuzzyTL structure. The importance of the adaptive methods depicted in Chapter 3 were clearly shown. Initially, a comparison was carried out between the full FuzzyTL framework, and the non-adapted systems using the inter contexts of the Intel laboratory dataset. A Wilcoxon signed-rank test was carried out between the non-adapted, and FuzzyTL frameworks. The use of the adaptive stages within the FuzzyTL framework were shown to reduce the errors in the output.

The similarity of input and output domain intervals was shown to have an impact on the performance of the non-adaptive system. The intersection of source and target input domain intervals, combined with the similarity of output interval domains, was shown to allow the non-adaptive system to achieve a high performance. This, however, was limited to specific domain cases.

A comparison was also made using the Robotics laboratory dataset. Similarly to the Intel dataset, the non-adaptive system was highlighted to fail to produce output when the input values stepped outside of the input interval domains. A Wilcoxon signed-rank test was carried out, highlighting that the two datasets came from different distributions. Further analysis showed that 82.5757% of contexts decreased in error through the application of the adaptive stages, substantiating the previous findings.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This chapter summarises the work within this thesis, by drawing together the hypotheses and discussing the key outcomes of the research. The major contributions are discussed along with future considerations for the work.

Two hypotheses were presented in Chapter 1. These were defined as:

Hypothesis 1 Where minimal unlabelled data is available within a target task, data in the form of a Transfer Learning (TL) process from contextually related but differing source tasks, can be used to learn predictive tasks.

Hypothesis 2 Adaptation of the transferred source domain through the use of unlabelled new data can increase the performance of Fuzzy Transfer Learning (FuzzyTL) in predicting target tasks.

Each of the hypotheses were tested through a series of experiments in Chapter 4. The experimentation confirmed both hypotheses. It was concluded that:

- The output of the *best* sample from the FuzzyTL framework was comparable in performance to a benchmark sample dataset, confirming the ability of the framework to use contextually related but different data to predict target tasks. This confirmed hypothesis 1.

- A comparative reduction in error was achieved, in the majority of contexts, when the adaptive processes were applied to the FuzzyTL framework. This confirmed hypothesis 2.

In the following sections, the findings of the experiments will be discussed further.

5.1.1 Contextually Differing Environments Can Act as Source Information

This thesis demonstrated that through the use of the FuzzyTL framework, only limited target data is necessary to predict an output. Using the methodology put forward in Chapter 3, the FuzzyTL was applied to a predictive learning task using limited unlabelled data. The experimental process focussed on dynamic environments that produce vague data. Intelligent Environments (IEs) epitomise this form. The modelling of IEs can be difficult. The unpredictable, real-world nature of such implementations, amplified by the addition of the human element, results in sporadic and uncertain data. The quantity, type and availability of data to model these applications can be an issue. Each situation is contextually different and constantly changing. Within many standard supervised learning strategies, training data must be labelled. This data is required to be in the same feature space and distribution as the target task data. This is often highly costly and time consuming to acquire. The criteria of certain implementations do not allow this structure to be produced. Unsupervised approaches can address this problem, however they are reliant on large quantities of unlabelled target data. Environments such as disaster recovery, environmental monitoring and specialised user groups can impact the quantity of the unlabelled data that can be sourced. This thesis focussed specifically on this problem domain.

In a simulated experimental set up, two Intelligent Environment (IE) datasets were used to test the hypotheses. Each dataset came from two contextually different environments. The first was the Intel Berkeley Laboratory in California, United States of America, the second from De Montfort University, Leicester, United Kingdom. The Intel Berkeley Laboratory dataset was composed of data readings from 54 environmental sensors. A subset of the sensors were used. Of the four parameters that the sensors captured, light and temperature were isolated. These were combined with time-stamp information. The De Montfort University dataset was produced from six sensors that recorded a combination of light and temperature. Again, a subset of the sensors were used. The environmental data was combined with temporal information. The performance of the FuzzyTL framework was measured using a Root Mean Squared Error (RMSE). This was based on a comparison of the FuzzyTL framework output against actual sensor readings. The framework used time and light readings to predict temperature values based on a model learnt from contextually different source data.

A main focus of this thesis was to understand whether the FuzzyTL framework can produce a predictive output based on little knowledge of the target domain. To contextualise the output of the framework, a benchmark dataset was produced. The benchmark was constructed by processing each of the datasets using the adapted Fuzzy Frequency Wang-Mendel (WM) process as described in Chapter 3, Section 3.4.2. The learning process was altered to become an Informed Supervised (IS) transfer learning approach. In IS transfer learning, labelled data is supplied to the learning process from both the target and source domains. A comparison was then made between the benchmark and the *best* FuzzyTL framework output. This comparison contextualised the output of the FuzzyTL framework.

To compare the datasets to the benchmark output, initially a paired t-test was used. Using the Intel Laboratory datasets, the FuzzyTL output and the benchmark were shown to be from different distributions. Closer inspection of the differences highlighted that in 24.4898% of cases of the FuzzyTL produced a lower RMSE value than the benchmark. Additionally, of the Intel Laboratory contexts analysed (2352 in total), 66.1990% contextually different source datasets produced an RMSE output that was equal, or within the minimum or maximum interval of the benchmark dataset. The analysis highlighted that the FuzzyTL framework was able to produce output that matched or surpassed the defined benchmark.

Overall, the Robotics Laboratory dataset substantiated the Intel Laboratory findings. Again a paired t-test was carried out to compare the best FuzzyTL output to the benchmark results. The two datasets were found to come from different distributions. Unlike the Intel Laboratory comparison, all of the contexts studied produced a higher RMSE when using contextually different source data. Drilling down into the data, however, showed that individual contextual instances produced RMSE values that were comparable to the benchmark.

The FuzzyTL was shown to be able to output predictive values using contextually different source data. Output from the FuzzyTL framework was comparable to a benchmark formed using target information. Although stronger within the Intel Laboratory dataset, the FuzzyTL framework was shown to produce predictive output across two differing real-world datasets.

5.1.2 Contextual Distance Has Little Effect on Error

To understand the impact of using contextually different source data, two separate context types were focussed upon: *inter* and *intra*. Chapter 4, Section 4.2.3 gives a detailed definition of each context. For the inter contexts, a Context Distance (CD) was defined (see Chapter 4, Section Section 4.2.3). Analysis was carried out on the relationship between the RMSE of the context, and the CD between the source and target contexts. The Intel Laboratory dataset showed that those contexts with the highest RMSE did not relate to data with the greatest CD. The same relationship occurred for the lowest RMSE contexts, the lowest CD did not produce the lowest RMSE. An examination of the data using a Pearson correlation showed no linear relationship. As the CD increases, the RMSE of the FuzzyTL framework remained within a similar distribution. The Robotics Laboratory dataset produced similar findings. Increases in the CD of data did not produce similar gain in the RMSE.

From these experiments, it was inferred that the extent of the CD has little impact on the quantity of error that is produced by the FuzzyTL framework. Further analysis on the Intel Laboratory dataset pointed towards the structure of the source and target data as causing change in the error produced. Correlation of the input values and the RMSE output showed there to be no relationship. However, a relationship was apparent between differences of source and target output values. The greater the distance between the consequent values, the larger the resulting RMSE. This was due to the structure of the framework. One element of the FuzzyTL framework is formed using a feedback system. The relationships between the input and output of the source data are mapped to the target data using the frameworks own output. As a result, larger initial differences in the output domains produced greater RMSE between the actual sensor readings and the framework output.

A similar assessment was made of the Robotics Laboratory dataset. The CD of each source and target context was compared to the RMSE output. Despite increases in the CD values, the RMSE values remained within a similar distribution. This substantiated the results of the Intel Laboratory dataset.

To assess the impact of context further, an analysis was made of intra contextual relationships. A full description of intra contexts can be found in Chapter 4, Section 4.2.3. The intra-contextual experiment used the Robotic Laboratory dataset as the source, and the Intel Laboratory dataset as the target. The performance of the FuzzyTL was again assessed using the RMSE calculated against the actual sensor output. The best output values were compared to those produced using inter contexts. Overall, the intra source data performed less well. The intra source data was unable to match the output of the inter source data. Despite this, individual intra contexts were shown to perform comparably to the inter contexts.

Overall, CD was judged to have a low impact on the resulting error produced by the FuzzyTL framework. Increases in RMSE were more closely associated with the proximity of the consequent (output) domains at the point of transfer. Further investigation into adaptive methods of the consequent domains are required. A possible avenue may come with the use of multiple source datasets (see Multiple Context Decision Making in Section 5.2).

5.1.3 Online Adaptation Decreases the Error of the FuzzyTL Output

The FuzzyTL framework is built upon the transfer and subsequent adaptation of source data. To investigate the impact of the adaptation process, and test the second hypothesis, a series of experiments based on a non-adapted version of the FuzzyTL framework were used. The Non-Adaptive (N-A) framework was constructed from a transferred Fuzzy Inference System (FIS). The learning process used to form the fuzzy system remained based on the FuzzyTL framework. The five stage online learning and adaptation process was removed. This gave a base framework for the comparison. Both the Intel Laboratory and Robotics Laboratory datasets were used within the experiments.

The Intel Laboratory dataset was initially analysed. A comparison was made between the performance of the N-A framework and the full FuzzyTL framework. The output of each context was assessed against the actual sensor readings. A RMSE of the results was produced. Each RMSE dataset was compared. The comparison was carried out using a Wilcoxon signed-rank test. This test showed that the N-A and fully adapted datasets came from significantly different distributions. Of the 2352 contexts compared, 2062 (87.6700%) produced a lower RMSE when the adaptive stages were incorporated. This clearly demonstrated that the introduction of the adaptive stages decreased the error produced. Failures in the N-A framework produced high RMSE values. The strict structure of the N-A framework resulted in failures. Target values that were beyond the source domain failed to produce an output. The adaptive process allowed the FuzzyTL to alter the domains to the target data, producing an output. In a minority of specific cases, the N-A frameworks out performed the full FuzzyTL framework. These special cases required the source and target interval domains to be in close proximity. Additionally, the target input domains were required to be proper subsets of the source input domains.

The same comparison was carried out on the Robotics Laboratory dataset. A Wilcoxon signed-rank test showed that the output from the full FuzzyTL framework and the non-adapted framework were from different distributions. Of the differences defined, 82.5757% (109 of the 132 contexts) produced a lower RMSE when the adaptive process was applied. Across the contexts, the largest reduction was $14.2349°\mathrm{C}$.

The inclusion of the adaptation process was shown to improve the performance of the FuzzyTL framework. A reduction in error between the actual sensor values and the output occurred in the majority of contexts. Where there was no improvement, the increase in error was marginal over the non-adapted framework.

5.1.4 Major Contributions

The approach to the complex, and uncertain problem domain that was set out in this thesis resulted in a number of significant contributions. Below, each of these contributions are set out and discussed:

A novel framework for the learning of target tasks using limited unlabelled target data and differing, related source labelled data. This thesis defined a novel framework for the learning of models to solve specific limited knowledge tasks. The basis of the problem focussed on environments were it is difficult, or in some situations, impossible to acquire training data. The framework was composed of a combination of Transfer Learning (TL) and Fuzzy Logic (FL) within a novel structure. The use of FL allows for the incorporation of approximation and a greater expressiveness of the uncertainty within the data. Using FL and a Fuzzy Inference System (FIS) as a base, TL is incorporated to dynamically model target tasks using contextually different source data.

A novel adaptive online learning methodology to enhance the transfer of FIS's between contextually differing learning tasks. The Fuzzy Transfer Learning (FuzzyTL) is composed of a transferred FIS system. The FIS acts as the basis for the learning of differing, but related tasks. To absorb the changes that occur between the source and target tasks, a five stage adaptation process was developed. The stages each constitute:

1. External adaptation of the input domains: Target input values that fall outside of the transferred source are used to adapt the FIS. Through this adaptation, knowledge of the target is absorbed into the source structure.

2. Internal adaptation of the input domains: Target input values that are contained within the transferred source domain intervals are used to adapt the transferred fuzzy sets.

3. Adaptation of the consequent domain: Incorporating the basic elements of TL, the third stage combines information from the source task and target task to adapt the FIS consequent values. By using the relationship of the labelled source data, and the unlabelled target data with the output of the FuzzyTL framework, a feedback system is adopted.

4. Rule modification through source comparison: The latter two stages approach the issues that arise with the transfer of a fuzzy rule base. The forth stage harnesses knowledge from the source data that may have been removed in the rule pruning process. The data is analysed to assess if it is applicable to the current target task.

5. Rule adaptation using Euclidean Distance measure: The final stage produces new rules from the source data. The process uses of a combined antecedent set extraction and euclidean distance approximation. This stage allows all target input values to produce an output.

A novel addition is provided to the Wang-Mendel (WM) method for the learning of fuzzy rules from numerical data using a fuzzy frequency approach. The automatic extraction of the fuzzy sets and rules in the FuzzyTL framework is based upon the use of the WM algorithm. Within this thesis, a novel extension of the WM method has been presented. In the standard WM process, the rule base is created by using the membership values of each data point. The full WM method is described in Chapter 2, Section 2.3.5.1. A data tuple made up of two inputs x_1, x_2 and a single output y each produces an output value based upon the largest membership in each set of the domain. Based on the membership values, the corresponding sets form a rule. Each data point, as a result, produces a single rule. To reduce the rule base, a pruning process is used based upon a weighted algorithm.

The new approach presented in this thesis expands the pruning process to incorporate a fuzzy frequency measure. The standard pruning method weights each rule based on its membership. The fuzzy frequency approach adds an additionally weighting based on the frequency that a rule occurred within the dataset. The weight is constructed using a fuzzy membership function. A full description of the fuzzy frequency approach is given in Chapter 3, Section 3.4.2. The combination of the original strength weighting and the fuzzy frequency impedes a single anomalous data point from having too much influence. Equally rules that are frequent but have have low strength will be equally impacted.

The first application of the Fuzzy Transfer Learning framework on Intelligent Environment (IE) datasets to perform predictive learning tasks. Within this thesis the FuzzyTL framework is shown to be able to output predictive values using source data from different contextual domains. The framework is applied to real-world IE datasets. Intelligent Environments (IEs) exemplify the structure and problem space which are the focus of this work. This is the first implementation of the FuzzyTL framework within IEs. The findings discussed in Chapter 4 show that the approach of the methodology is applicable to these highly dynamic and uncertain environments.

5.2 Recommendations and Future Work

In the following sections, a number of recommendations and possible future work are put forward.

Comparison of Wang-Mendel Method to Other Rule Generation Methodologies Within the FuzzyTL framework, a WM methodology was used to extract a FIS using numerical data. As the focus of this research was to investigate the use of transferred information, and its impact on learning using a FIS, only a preliminary comparison of FIS production methods was carried out. The production of a FIS can follow a number of routes. Within this thesis, a study was carried out investigating varying forms of inductive methods to produce both fuzzy sets and fuzzy rules. Further methods exist outside of this study, though the comparison of these was outside the scope of this thesis. An in depth comparison of extraction methodologies would highlight other applicable methodologies. This may allow for the further extension of the FuzzyTL framework.

The extension of the WM methodology is also of interest. This thesis offered an addition to the WM approach, however, a particular area of expansion is the use of varying methods such as Genetic Algorithms (GAs) (Casillas et al. 2000) and Particle Swarm Optimisation (PSO) (Oliveira Costa et al. 2011) to optimise the rule base following the initial rule extraction. There is scope to investigate the impact of a highly optimised source FIS on the target task.

Automation of Set Extraction From Data The current FuzzyTL framework uses a simple system to define the number of sets that are used within the FIS. This is based on a preassigned value. Previous research has been carried out into the extraction of both fuzzy sets and fuzzy rules from labelled data. As discussed within Chapter 2, fuzzy clustering methods use patterns contained within labelled data to extract suitable sets. Fuzzy clustering, however, also requires a predefined number of sets to initiate the process. A number of methods have been proposed to overcome this. The fuzzy clustering algorithm can be initialised with an overestimation of the required number of clusters (Setnes 2000). A higher possibility is then produced that the important regions of the domain are covered. Less important, and redundant clusters can then be removed to extract fuzzy sets. Of interest is the application of such methods to automate the process of assigning set quantities. The removal of the need to use expert knowledge to assign set quantities would further automate the framework. This area of research would extend the data driven nature of the framework structure.

Use of Multiple Context Data The composition of the source data has been shown to have a direct effect on the outcome of the predictive value of the FuzzyTL framework. The proximity of the source and target domains can have a direct effect on the error produced. Certain conditions of a source domain can have adverse affects on the learning of the target domain. Anomalous or erroneous data can produce a model that is incorrect. To tackle this issue, research has been conducted into the use of multiple source domains within TL (Luo et al. 2008, Yao & Doretto 2010). The use of multiple sources can increase the chance of discovering a source domain that is close to the target. The extension of the FuzzyTL framework to incorporate multiple source data may reduce the impact of negative transfer. A further extension of this concept is the use of multiple co-operative contextual decision making. The concept of multiple co-operative source data extends the possible boost gained from a single data source by using a collection. Through a decision making process, the most eligible information is transferred to the target task based on the source data available.

The use of multiple source contexts, can not only assist in increasing the performance of the FuzzyTL framework, but also consolidate results that have been attained. Previous work carried out by the author has investigated the use of multiple sensor information within a Wireless Sensor Network (WSN) to identify anomalous readings (Shell et al. 2010). Extension of this work through the incorporation of multiple source domains may allow for the application of anomaly detection.

Implementation of FuzzyTL to other applications The application of the FuzzyTL framework in Chapter 4 to IE datasets has proven that it is applicable to predictive, real-world tasks. The uncertain and dynamic structure of the data has parallels to other real-world applications. The broad nature of the learning attributes allow the framework to be applicable to many situations.

There is scope to investigate the scalability of the framework under increasing levels of complexity, and to investigate the use of different data structures. Specific situations of interest include disaster situations were little or no data of the current situation maybe known, although previous knowledge is accessible, isolated Natural Language Processing (NLP) contexts were no or few examples exist to train from, and groups were information is extremely difficult to ascertain such as disabled users.

A preliminary work has been undertaken in the area of eye-gaze gesture recognition (Shell et al. 2012) by the author to recognise gestures of disabled users using non-disabled source data. This is an emerging research topic that can allow the further development of the FuzzyTL framework.

Bibliography

Acampora, G., Gaeta, M., Loia, V. & Vasilakos, A. V. (2010), 'Interoperable and adaptive fuzzy services for ambient intelligence applications', *ACM Trans. Auton. Adapt. Syst.* **5**(2), 8:1–8:26.

Adriaenssens, V., De Baets, B., Goethals, P. & De Pauw, N. (2004), 'Fuzzy rule-based models for decision support in ecosystem management', *Science of the Total Environment* **319**(1), 1–12.

Akhlaghinia, M., Lotfi, A. & Langensiepen, C. (2007), Soft computing prediction techniques in ambient intelligence environments, *in* 'Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International', pp. 1 –6.

Anderson, J. & Davis, J. (1995), *An introduction to neural networks*, Vol. 1, MIT Press.

Angelov, P. & Buswell, R. (2001), Evolving rule-based models: A tool for intelligent adaptation, *in* 'IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th', Vol. 2, pp. 1062 –1067 vol.2.

Angelov, P. & Kordon, A. (2010), 'Adaptive inferential sensors based on evolving fuzzy models', *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **40**(2), 529–539.

Angelov, P. & Zhou, X. (2006), Evolving fuzzy systems from data streams in real-time, *in* 'Evolving Fuzzy Systems, 2006 International Symposium on', pp. 29 –35.

Arnold, A., Nallapati, R. & Cohen, W. (2007), A comparative study of methods for transductive transfer learning, *in* 'Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on', pp. 77 –82.

Barrett, S., Taylor, M. & Stone, P. (2010), Transfer learning for reinforcement learning on a physical robot, *in* 'Ninth International Conference on Autonomous Agents and Multiagent Systems-Adaptive Learning Agents Workshop (AAMAS-ALA)'.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A. & Riboni, D. (2010), ‘A survey of context modelling and reasoning techniques’, *Pervasive and Mobile Computing* **6**(2), 161–180.

Bezdek, J. (1981), *Pattern recognition with fuzzy objective function algorithms*, Kluwer Academic Publishers Norwell, MA.

Bezdek, J. C. (1973), Fuzzy Mathematics in Pattern Classification, PhD thesis, Applied Math. Center, Cornell University, Ithaca.

Bezdek, J., Ehrlich, R. & Full, W. (1984), ‘Fcm: The fuzzy c-means clustering algorithm’, *Computers & Geosciences* **10**(2), 191–203.

Black, M. (1937), ‘Vagueness. an exercise in logical analysis’, *Philosophy of Science 4* pp. 425–455.

Bonarini, A. (1997), ‘Anytime learning and adaptation of structured fuzzy behaviors’, *Adaptive Behavior* **5**(3-4), 281–315.

Cabrita, C., Botzheim, J., Gedeon, T., Ruano, A., Kóczy, L. & Fonseca, C. (2006), Bacterial memetic algorithm for fuzzy rule base optimization, *in* ‘Automation Congress, 2006. WAC’06. World’, pp. 1–6.

Cao, B., Pan, S., Zhang, Y., Yeung, D. & Yang, Q. (2010), Adaptive transfer learning, *in* ‘Proceedings of the 24th AAAI Conference on Artificial Intelligence’, pp. 407–412.

Caruana, R. (1997), ‘Multitask learning’, *Mach. Learn.* **28**, 41–75.

Casillas, J., Cordón, O. & Herrera, F. (2000), Improving the wang and mendels fuzzy rule learning method by inducing cooperation among rules, *in* ‘Proceedings of the 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference’, pp. 1682–1688.

Castro, J., Castro-Schez, J. & Zurita, J. (1999), ‘Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems’, *Fuzzy sets and systems* **101**(3), 331–342.

Cesa-Bianchi, N., Conconi, A. & Gentile, C. (2004), ‘On the generalization ability of on-line learning algorithms’, *Information Theory, IEEE Transactions on* **50**(9), 2050–2057.

Chapelle, O., Schölkopf, B., Zien, A. et al. (2006), *Semi-supervised learning*, Vol. 2, MIT press Cambridge, MA:.

Chen, M. & Linkens, D. (2001), Rule-base self-generation and simplification for data-driven fuzzy models, *in* 'Fuzzy Systems, 2001. The 10th IEEE International Conference on', Vol. 1, pp. 424–427.

Chen, M. & Linkens, D. (2004), 'Rule-base self-generation and simplification for data-driven fuzzy models', *Fuzzy sets and systems* **142**(2), 243–265.

Collobert, R. & Weston, J. (2008), A unified architecture for natural language processing: Deep neural networks with multitask learning, *in* 'Proceedings of the 25th international conference on Machine learning', pp. 160–167.

Cook, D. & Das, S. (2004), *Smart Environments: Technology, Protocols and Applications*, Wiley Series on Parallel and Distributed Computing, John Wiley & Sons, Hoboken, New Jersey.

Cook, D. & Das, S. (2007), 'How smart are our environments? an updated look at the state of the art', *Pervasive and Mobile Computing* **3**(2), 53–73.

Cook, D., Feuz, K. & Krishnan, N. (2012), 'Transfer learning for activity recognition: A survey', http://www.eecs.wsu.edu/ cook/pubs/kais12.pdf. Accessed 15th November, 2012.

Copetti, A., Loques, O., Leite, J., Barbosa, T. & da Nobrega, A. (2009), Intelligent context-aware monitoring of hypertensive patients, *in* 'Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on', pp. 1–6.

Cormier, S. & Hagman, J. (1987), *Transfer of learning: Contemporary research and applications.*, Academic Press, Waltham, Massachusetts.

Dai, W., Yang, Q., Xue, G. & Yu, Y. (2007), Boosting for transfer learning, *in* 'Proceedings of the 24th international conference on Machine learning', pp. 193–200.

Damin-Reyes, P., Favela, J. & Contreras-Castillo, J. (2011), 'Uncertainty management in context-aware applications: Increasing usability and user trust', *Wireless Personal Communications* **56**, 37–53.

Daumé III, H. & Marcu, D. (2006), 'Domain adaptation for statistical classifiers', *Journal of Artificial Intelligence Research* **26**(1), 101–126.

Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J. & Hong, W. (2004), Model-driven data acquisition in sensor networks, *in* 'Proceedings of the Thirtieth international conference on Very large data bases-Volume 30', pp. 588–599.

Dey, A. (2001), 'Understanding and using context', *Personal and ubiquitous computing* **5**(1), 4–7.

Dey, A., Mankoff, J. & Abowd, G. (2000), Distributed mediation of imperfectly sensed context in aware environments, Technical report.

Doctor, F., Hagras, H. & Callaghan, V. (2005), 'A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments', *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* **35**(1), 55–65.

Dourish, P. (2004), 'What we talk about when we talk about context', *Personal Ubiquitous Comput.* **8**(1), 19–30.

Doyle, J. (1979), 'A truth maintenance system', *Artificial intelligence* **12**(3), 231–272.

Dunn, J. (1973), 'A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters', *Journal of Cybernetics* **3**, 32–57.

Elnahrawy, E. & Nath, B. (2004), 'Context-aware sensors', *Wireless Sensor Networks* **2920**, 77–93.

Erkan, G., Ozgur, A. & Radev, D. (2007), Semi-supervised classification for extracting protein interaction sentences using dependency parsing, *in* 'Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)', Vol. 1, pp. 228–237.

Fazel Zarandi, M., Rezaee, B., Turksen, I. & Neshat, E. (2009), 'A type-2 fuzzy rule-based expert system model for stock price analysis', *Expert Systems with Applications* **36**(1), 139–154.

Filev, D. & Tseng, F. (2006), Novelty detection based machine health prognostics, *in* 'Evolving Fuzzy Systems, 2006 International Symposium on', pp. 193–199.

Fleming, P. & Purshouse, R. (2002), 'Evolutionary algorithms in control systems engineering: a survey', *Control engineering practice* **10**(11), 1223–1241.

Fodor, I. (2002), 'A survey of dimension reduction techniques', *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory* **9**, 1–18.

Gal, L., Botzheim, J. & Kóczy, L. (2008), Improvements to the bacterial memetic algorithm used for fuzzy rule base extraction, *in* 'Computational Intelligence for Measurement Systems and Applications, 2008. CIMSA 2008. 2008 IEEE International Conference on', pp. 38–43.

Gath, I. & Geva, A. (1989), 'Unsupervised optimal fuzzy clustering', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **11**(7), 773 –780.

Genco, A. & Sorce, S. (2010), *Pervasive Systems and Ubiquitous Computing*, Wit Press, Southampton.

George, S. M., Zhou, W., Chenji, H., Won, M., Lee, Y. O., Pazarloglou, A., Stoleru, R. & Barooah, P. (2010), 'Distressnet: a wireless ad hoc and sensor network architecture for situation management in disaster response', *Communications Magazine, IEEE* **48**(3), 128–136.

Gorski, N. & Laird, J. (2006), Experiments in transfer across multiple learning mechanisms, *in* 'Proceedings of the ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning'.

Grefenstette, J. (1986), 'Optimization of control parameters for genetic algorithms', *Systems, Man and Cybernetics, IEEE Transactions on* **16**(1), 122 –128.

Guillaume, S. (2001), 'Designing fuzzy inference systems from data: An interpretability-oriented review', *Fuzzy Systems, IEEE Transactions on* **9**(3), 426 –443.

Hagras, H. (2007), 'Type-2 flcs: A new generation of fuzzy controllers', *Computational Intelligence Magazine, IEEE* **2**(1), 30 –43.

Hagras, H., Callaghan, V. & Colley, M. (2004), 'Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online fuzzy–genetic system', *Fuzzy Sets and Systems* **141**(1), 107–160.

Hagras, H., Callaghan, V., Colley, M. & Clarke, G. (2003), 'A hierarchical fuzzy–genetic multi-agent architecture for intelligent buildings online learning, adaptation and control', *Information Sciences* **150**(1), 33–57.

Hagras, H., Doctor, F., Callaghan, V. & Lopez, A. (2007), 'An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments', *Fuzzy Systems, IEEE Transactions on* **15**(1), 41 –55.

Hong, T. & Lee, C. (1996), 'Induction of fuzzy rules and membership functions from training examples', *Fuzzy Sets and Systems* **84**(1), 33–47.

Hu, D. & Yang, Q. (2011), Transfer learning for activity recognition via sensor mapping, *in* 'Twenty-Second International Joint Conference on Artificial Intelligence', pp. 1962–1967.

Ishibuchi, H., Nozaki, K., Tanaka, H., Hosaka, Y. & Matsuda, M. (1994), 'Empirical study on learning in fuzzy systems by rice taste analysis', *Fuzzy sets and systems* **64**(2), 129–144.

Ishibuchi, H. & Yamamoto, T. (2004), ‘Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining’, *Fuzzy Sets and Systems* **141**(1), 59–88.

Jang, J.-S. (1997), *Fuzzy inference systems*, Prentice-Hall, Upper Saddle River, New Jersey.

Jang, S. (2005), ‘Implementation of Context-Aware Application Model in Smart Environments’, *Gwangju Institute of Science and Technology, PhD thesis* .

Jung, Y., Lee, Y., Lee, D., Ryu, K. & Nittel, S. (2008), ‘Air Pollution Monitoring System based on Geosensor Network’, **3**, 1370–1373.

Kaelbling, L. P., Littman, M. L. & Moore, A. W. (1996), ‘Reinforcement learning: A survey’, *Journal of Artificial Intelligence Research* **4**, 237–285.

Kasabov, N. (1996), ‘Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems’, *Fuzzy sets and Systems* **82**(2), 135–149.

Kennedy, J. & Eberhart, R. (1995), Particle swarm optimization, *in* ‘Neural Networks, 1995. Proceedings., IEEE International Conference on’, Vol. 4, pp. 1942–1948.

Klir, G., St Clair, U. & Yuan, B. (1997), *Fuzzy set theory: foundations and applications*, Prentice-Hall, Inc, New Jersey.

Klir, G. & Wierman, M. (1999), *Uncertainty-based information: Elements of generalized information theory*, Vol. 15, Springer Verlag, New York.

Larochelle, H., Erhan, D. & Bengio, Y. (2008), Zero-data learning of new tasks, *in* ‘AAAI Conference on Artificial Intelligence’, Vol. 1, pp. 2–2.

Leberman, S., McDonald, L. & Doyle, S. (2006), *The transfer of learning: Participants’ perspectives of adult education and training*, Gower Publishing Company, Limited, Aldershot.

Lee, C. (1990), ‘Fuzzy logic in control systems: fuzzy logic controller. i’, *Systems, Man and Cybernetics, IEEE Transactions on* **20**(2), 404–418.

Leng, G., McGinnity, T. & Prasad, G. (2005), ‘An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network’, *Fuzzy Sets and Systems* **150**(2), 211–243.

Liu, Q., Liao, X., Li, H., Stack, J. & Carin, L. (2009), ‘Semisupervised multitask learning’, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **31**(6), 1074 –1086.

Luo, C., Ji, Y., Dai, X. & Chen, J. (2012), Active learning with transfer learning, *in* 'Proceedings of the 2012 Student Research Workshop', pp. 13–18.

Luo, P., Zhuang, F., Xiong, H., Xiong, Y. & He, Q. (2008), Transfer learning from multiple source domains via consensus regularization, *in* 'Proceedings of the 17th ACM conference on Information and knowledge management', CIKM '08, ACM, New York, NY, USA, pp. 103–112.

Macaulay, C. (2001), *Transfer of learning*, Vocational Education, London.

Madden, S. (2004), 'Intel lab data', http://db.csail.mit.edu/labdata/labdata.html. Published on 2nd June 2004.

Mamdani, E. (1974), 'Application of fuzzy algorithms for control of simple dynamic plant', *Electrical Engineers, Proceedings of the Institution of* **121**(12), 1585 –1588.

Martin Larsen, P. . (1980), 'Industrial applications of fuzzy logic control', *International Journal of Man-Machine Studies* **12**(1), 3–10.

McKeough, A., Lupart, J. & Marini, A. (1995), *Teaching for transfer: Fostering generalization in learning*, Lawrence Erlbaum, New Jersey.

Mendel, J. (1995), 'Fuzzy logic systems for engineering: a tutorial', *Proceedings of the IEEE* **83**(3), 345–377.

Mendel, J. (2000), 'Uncertainty, fuzzy logic, and signal processing', *Signal Processing* **80**(6), 913–933.

Mendel, J. & John, R. (2002), 'Type-2 fuzzy sets made simple', *IEEE Transactions on Fuzzy Systems* **10**(2), 117–127.

Meyer, S. & Rakotonirainy, A. (2003), A survey of research on context-aware homes, *in* 'Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21', pp. 159–168.

Mihalkova, L., Huynh, T. & Mooney, R. (2007), Mapping and revising markov logic networks for transfer learning, *in* 'Proceedings of the national conference on artificial intelligence', Vol. 22, p. 608.

Miller, E. (2002), Learning from one example in machine vision by sharing probability densities, PhD thesis, Yale University.

Moallem, P., Mousavi, B. & Monadjemi, S. (2011), 'A novel fuzzy rule base system for pose independent faces detection', *Applied Soft Computing* **11**(2), 1801–1810.

Moore, R. (1987), *Methods and applications of interval analysis*, Vol. 2, Society for Industrial Mathematics, Philadelophia.

Mozer, M. C. (1998), The neural network house: An environment hat adapts to its inhabitants, *in* 'Proc. AAAI Spring Symp. Intelligent Environments'.

Nauck, D. (1997), Neuro-fuzzy systems: Review and prospects, *in* 'In Proceedings of Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT97)', pp. 1044–1053.

Nozaki, K., Ishibuchi, H. & Tanaka, H. (1997), 'A simple but powerful heuristic method for generating fuzzy rules from numerical data', *Fuzzy Sets and Systems* **86**(3), 251–270.

Oliveira Costa, S., Nedjah, N. & De Macedo Mourelle, L. (2011), 'Automatic adaptive modeling of fuzzy systems using particle swarm optimization', *Transactions on computational science VIII* pp. 71–84.

Pan, S., Kwok, J. & Yang, Q. (2008), Transfer learning via dimensionality reduction, *in* 'Proceedings of the 23rd national conference on Artificial intelligence', pp. 677–682.

Pan, S. & Yang, Q. (2009), 'A survey on transfer learning', *IEEE Transactions on Knowledge and Data Engineering* pp. 1345–1359.

Perkins, D. & Salomon, G. (1992), 'Transfer of learning', *International encyclopedia of education* **2**, 1.

Puteh, S., Langensiepen, C. & Lotfi, A. (2012), Fuzzy ambient intelligence for intelligent office environments, *in* 'Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on', pp. 1 –6.

Raina, R., Battle, A., Lee, H., Packer, B. & Ng, A. (2007), Self-taught learning: transfer learning from unlabeled data, *in* 'Proceedings of the 24th international conference on Machine learning', pp. 759–766.

Ranganathan, A., Al-Muhtadi, J. & Campbell, R. (2004), 'Reasoning about uncertain contexts in pervasive computing environments', *Pervasive Computing, IEEE* **3**(2), 62–70.

Roychowdhury, S. & Pedrycz, W. (2001), 'A survey of defuzzification strategies', *International Journal of intelligent systems* **16**(6), 679–695.

Rutishauser, U., Joller, J. & Douglas, R. (2005), 'Control and learning of ambience by an intelligent building', *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* **35**(1), 121 – 132.

Saha, D. & Mukherjee, A. (2003), 'Pervasive computing: a paradigm for the 21st century', *Computer* **36**(3), 25–31.

Satyanarayanan, M. (2001), 'Pervasive computing: Vision and challenges', *Personal Communications, IEEE* **8**(4), 10–17.

Schilit, B., Adams, N. & Want, R. (1994), Context-Aware Computing Applications, *in* 'Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications-Volume 00', pp. 85–90.

Scott, J., Krumm, J., Meyers, B., Brush, A. & Kapoor, A. (2010), Home heating using gps-based arrival prediction, Technical report, Tech. rep., Microsoft Research, USA.

Sengupta, A. & Pal, T. (2000), 'On comparing interval numbers', *European Journal of Operational Research* **127**(1), 28–43.

Setnes, M. (2000), 'Supervised fuzzy clustering for rule extraction', *Fuzzy Systems, IEEE Transactions on* **8**(4), 416–424.

Setnes, M. & Roubos, H. (2000), 'Ga-fuzzy modeling and classification: complexity and performance', *Fuzzy Systems, IEEE Transactions on* **8**(5), 509–522.

Shell, J. & Coupland, S. (2012), 'Towards fuzzy transfer learning for intelligent environments', *Ambient Intelligence* pp. 145–160.

Shell, J., Coupland, S. & Goodyer, E. (2010), Fuzzy data fusion for fault detection in Wireless Sensor Networks, *in* 'Computational Intelligence (UKCI), 2010 UK Workshop on', pp. 1–6.

Shell, J., Vickers, S., Coupland, S. & Istance, H. (2012), Towards dynamic accessibility through soft gaze gesture recognition, *in* 'Computational Intelligence (UKCI), 2012 12th UK Workshop on', pp. 1–8.

Sudkamp, T. & Hammell III, R. (1994), 'Interpolation, completion, and learning fuzzy rules', *Systems, Man and Cybernetics, IEEE Transactions on* **24**(2), 332–342.

Sugeno, M. & Yasukawa, T. (1993), 'A fuzzy-logic-based approach to qualitative modeling', *Fuzzy Systems, IEEE Transactions on* **1**(1), 7.

Sutton, R. & Barto, A. (1998), *Reinforcement learning: An introduction*, Vol. 1, Cambridge Univ Press.

Talvitie, E. & Singh, S. (2007), An experts algorithm for transfer learning, *in* 'Proceedings of the Twentieth International Joint Conference on Artificial Intelligence', pp. 1065–1070.

Taylor, M., Whiteson, S. & Stone, P. (2006), Transfer learning for policy search methods, *in* 'ICML Workshop on Structural Knowledge Transfer for Machine Learning'.

Teodorovic, D., Lucic, P., Popovic, J., Kikuchi, S. & Stanic, B. (2001), Intelligent isolated intersection, *in* 'Fuzzy Systems, 2001. The 10th IEEE International Conference on', Vol. 1, pp. 276–279.

The Sensor Network Museum (2012), Internet. Accessed 27th December 2012 : http://www.snm.ethz.ch/snmwiki/Projects/Mica2Dot.

Thrun, S. (1996), Is learning the n-th thing any easier than learning the first?, *in* 'Advances in Neural Information Processing Systems', pp. 640–646.

Thrun, S. & Mitchell, T. (1995), 'Lifelong robot learning* 1', *Robotics and autonomous systems* **15**(1-2), 25–46.

Torrey, L. & Shavlik, J. (2009), 'Transfer learning', *Handbook of Research on Machine Learning Applications. IGI Global* **3**, 17–35.

Wagner, C. & Hagras, H. (2010), An approach for the generation and adaptation of zslices based general type-2 fuzzy sets from interval type-2 fuzzy sets to model agreement with application to intelligent environments, *in* 'Fuzzy Systems (FUZZ), 2010 IEEE International Conference on', pp. 1 –8.

Wang, L. (1999), *A Course on Fuzzy Systems*, Prentice-Hall press, USA.

Wang, L. & Mendel, J. (1992), 'Generating fuzzy rules by learning from examples', *Systems, Man and Cybernetics, IEEE Transactions on* **22**(6), 1414–1427.

Wang, L.-X. (2003), 'The wm method completed: a flexible fuzzy system approach to data mining', *Fuzzy Systems, IEEE Transactions on* **11**(6), 768 – 782.

Weiser, M. (1991), 'The computer for the 21st century', *Scientific American* **265**(3), 94–104.

Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J. & Welsh, M. (2005), Monitoring volcanic eruptions with a wireless sensor network, *in* 'Proc. Second European Workshop on Wireless Sensor Networks (EWSN05)'.

Wu, S., Er, M. J. & Gao, Y. (2001), ‘A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks’, *Fuzzy Systems, IEEE Transactions on* **9**(4), 578 –594.

Xu, Q. & Yang, Q. (2011), ‘A survey of transfer and multitask learning in bioinformatics’, *Journal of Computing Science and Engineering* **5**(3), 257–268.

Yang, M. (1993), ‘A survey of fuzzy clustering’, *Mathematical and Computer modelling* **18**(11), 1–16.

Yang, X., Yuan, J., Yuan, J. & Mao, H. (2010), ‘An improved wm method based on pso for electric load forecasting’, *Expert Systems with Applications* **37**(12), 8036–8041.

Yao, Y. & Doretto, G. (2010), Boosting for transfer learning with multiple sources, *in* ‘Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on’, pp. 1855–1862.

Youngblood, G., Heierman, E., Holder, L. & Cook, D. (2005), Automation intelligence for the smart environment, *in* ‘International Joint Conference On Artificial Intelligence’, Vol. 19, p. 1513.

Zadeh, L. (1965), ‘Fuzzy sets*’, *Information and control* **8**(3), 338–353.

Zadeh, L. (1973), ‘Outline of a new approach to the analysis of complex systems and decision processes’, *Systems, Man and Cybernetics, IEEE Transactions on* (1), 28–44.

Zadeh, L. (1988), ‘Fuzzy logic’, *Computer* **21**(4), 83–93.

Zadeh, L. (1994), ‘Soft computing and fuzzy logic’, *Software, IEEE* **11**(6), 48 –56.

Zhou, X. & Angel, P. (2006), Real-time joint landmark recognition and classifier generation by an evolving fuzzy system, *in* ‘Fuzzy Systems, 2006 IEEE International Conference on’, pp. 1205–1212.

Zhu, X. (2006), ‘Semi-supervised learning literature survey’, *Computer Science, University of Wisconsin-Madison* .

Zhu, X., Ghahramani, Z., Lafferty, J. et al. (2003), Semi-supervised learning using gaussian fields and harmonic functions, *in* ‘MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-’, Vol. 20, p. 912.

Zhu, X. & Goldberg, A. (2009), ‘Introduction to semi-supervised learning’, *Synthesis lectures on artificial intelligence and machine learning* **3**(1), 1–130.

Appendix

Published Papers

Below is a list of the papers produced during the period of this research. Paper three discusses the application of the Fuzzy Transfer Learning (FuzzyTL) framework in the area of gaze gesture recognition. Paper four demonstrates a version of the methodology presented in this thesis using the Intel Laboratory dataset.

1. J. Shell, S. Coupland, and E. Goodyer. "Fuzzy data fusion for fault detection in Wireless Sensor Networks". In: *Computational Intelligence (UKCI), 2010 UK Workshop on.* 2010, pp. 1–6

2. J. Shell and S. Coupland. "Improved Decision Making Using Fuzzy Temporal Relationships within Intelligent Assisted Living Environments". In: *Intelligent Environments (IE), 2011 7th International Conference on.* 2011, pp. 149–156

3. J. Shell et al. "Towards dynamic accessibility through soft gaze gesture recognition". In: *Computational Intelligence (UKCI), 2012 12th UK Workshop on.* 2012, pp. 1–8

4. J. Shell and S. Coupland. "Towards Fuzzy Transfer Learning for Intelligent Environments". In: *Ambient Intelligence* (2012), pp. 145–160