Identification and Prediction of Abnormal Behaviour Activities of Daily Living in Intelligent Environments

SAWSAN MOUSA MAHMOUD

A thesis submitted in partial fulfilment of the requirements of
Nottingham Trent University for the degree of

Doctor of Philosophy

May 2012

This thesis is dedicated to my parents with great gratitude. Undoubtedly, without their prayers and support this thesis would have been impossible.

Acknowledgements

The work described in this thesis was carried out at Nottingham Trent University between March 2008 and March 2012, while the researcher was working as a full-time doctoral research student.

I owe my utmost sincere gratitude to my honourable director of studies, Dr. Ahmad Lotfi, for his guidance over the last four years. His office was always open for me for any inquiry despite his many responsibilities. Also, his relentless effort and extensive suggestions around my work were indispensable to the accomplishment of this research. I would like to extend a huge thank to Dr. Caroline Langensiepen for her advice, support, feedback and knowledge throughout the development of my thesis. I would like also to thank other members of my supervisory team, Prof. Nasser Sherkat, and Dr. Taha Osman, for their support throughout the course of this project work.

Moreover, I want really to show appreciation to my sponsor, Ministry of Higher Education of Iraq for awarding me this Scholarship. Without this precious opportunity, the graduate study would have been impossible. The financial support of School of Science and Technology at Nottingham Trent University to participate in various international conferences is also most appreciated. The data sets provided by Just Checking Ltd. and utilised in this research is also gratefully acknowledged.

Thanks are extended to my extended family, parents, sisters, brothers and uncles. Their prayers and faith have inspired and encouraged me to continue with this study. In particular, I owe a huge debt of gratitude to my sister Batoul and her family for being my guarantor underwriting contract with my sponsor. Thanks also should gone to friends and colleagues in the School of Science and Technology in Nottingham Trent University. In addition, I would like to pay tribute to my family for their support, sympathy, and patience throughout the journey of this study. In particular, I would like to say thanks to my husband Ali and our lovely daughters, Sarah and Saraa. Your joy and love have always put me up during the well-being and sadness of the last four years.

Sawsan Mousa Mahmoud
March 2012

Abstract

The aim of this research is to investigate efficient mining of useful information from a sensor network forming an Ambient Intelligence (AmI) environment. In this thesis, we investigate methods for supporting independent living of the elderly (and specifically patients who are suffering from dementia) by means of equipping their home with a simple sensor network to monitor their behaviour and identify their Activities of Daily Living (ADL). Dementia is considered to be one of the most important causes of disability in the elderly. Most patients would prefer to use non-intrusive technology to help them to maintain their independence. Such monitoring and prediction would allow the caregiver to see any trend in the behaviour of the elderly person and to be informed of any abnormal behaviour.

Employing a sensor network system allows us to extract daily behavioural patterns of the occupant in an Intelligent Inhabited Environment (IIE). This information is then used to build a behavioural model of the occupant which ultimately is applied to predict the future values representing the expected occupancy in the monitored environment. Challenges of employing wired and wireless sensor network have been widely researched. However, pattern analysis and prediction of sensory data is becoming an increasing scientific challenge and this research investigates appropriate means of pattern mining and prediction within the IIE.

Door entry and occupancy sensors are used to extract the movement patterns of the occupant. These sensors produce long sequences of data as binary time series, indicating presence or absence of the occupant in different areas. It is essential to convert these binary series

into a more flexible and efficient format before they are processed for any further analysis and prediction. Different ways of representing and visualizing the large sensor data sets in a format suitable for predicting and identifying the behaviour patterns are investigated.

A two-stage integration of Principal Component Analysis (PCA) and Fuzzy Rule-Based System (FRBS) is proposed to identify important information regarding outliers or abnormal behaviours in ADLs. In the first stage, binary dissimilarities or distance measures are used to measure the distances between the activities. PCA is then applied to find two indices of Hotelling's T^2 and Squared Prediction Error (SPE). In the second stage of the process, the calculated indices are provided as inputs to FRBSs to model them heuristically. They are used to identify outliers and classify them. The proposed system identifies user activities and helps in distinguishing between the normal and abnormal behavioural patterns of the ADLs.

Data provided for this investigation was from real environments and from a previously developed simulator. The simulator was modified to include trending behaviour in the activities of daily living. Therefore, in the occupancy signal generated by the simulator, both seasonality and trend are included in occupant's movements. Prediction models are built through Recurrent Neural Networks (RNN) after converting the occupancy binary time series. RNN have shown a great ability in finding the temporal relationships of input patterns. In this thesis, RNN are compared to evaluate their abilities to accurately predict the behaviour patterns. The experimental results show that Echo State Network (ESN) and Non-linear Autoregressive netwoRk with eXogenous (NARX) inputs correctly extract the long term prediction patterns of the occupant and outperformed the classical Elman network.

Publications

The following publications have been published as a direct result of this thesis:

Refereed Journal Papers

Sawsan M. Mahmoud, Ahmad Lotfi, Caroline Langensiepen, Behavioural Pattern Identification and Prediction in Intelligent Environments, Journal of Applied Soft Computing, (Submitted).

Ahmad Lotfi, Caroline Langensiepen, Sawsan M. Mahmoud and M. J. Akhlaghinia, Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour, Journal of Ambient Intelligent Human Computing, Springer Berlin/Heidelberg, pp. 1-14, DOI: 10.1007/s12652-010-0043-x, 2011.

Refereed Conference Papers

Sawsan M. Mahmoud, Ahmad Lotfi and Caroline Langensiepen, User Activities Outlier Detection System Using Principal Component Analysis and Fuzzy Rule-Based System, Fifth International Conference on PErvasive Technologies Related to Assistive Environments, Crete, Greece, ISBN 978-1-4503-1300-1/12/06, 2012.

Taha Osman, Suganth Ramaswamy, Sawsan Mahmoud and Mahmoud Saeed, Modelling Decision Support Systems for Remote Dementia Care: A Semantic Web Approach, Semantic Interoperability in Medical Informatics, Heraklion (Crete), Greece, May 27, 2012.

Sawsan M. Mahmoud, Ahmad Lotfi and Caroline Langensiepen, Behavioural Pattern Identification in a Smart Home using Binary Similarity and Dissimilarity Measures, Seventh International Conference on Intelligent Environments, Nottingham, UK, pp. 55-60, DOI: 10.1109/IE.2011.53, 2011.

Sawsan M. Mahmoud, Ahmad Lotfi and Caroline Langensiepen, Abnormal Behaviours Identification for an Elders Life Activities using Dissimilarity Measurements, 4th International Conference on PErvasive Technologies Related to Assistive Environments, Crete, Greece, DOI: 10.1145/2141622.2141653, 2011.

Sawsan M. Mahmoud, M. Javad Akhlaghinia, Ahmad Lotfi and Caroline Langensiepen, Trend Modelling of Elderly Lifestyle within an Occupancy Simulator, UKSim 13th International Conference on Modelling and Simulation, Cambridge, UK, pp. 156-161, DOI: 10.1109/UKSIM.2011.38, 2011.

Sawsan M. Mahmoud, Ahmad Lotfi and Caroline Langensiepen, Occupancy Pattern Extraction and Prediction in an Inhabited Intelligent Environment using NARX Networks, Sixth International Conference on Intelligent Environments, Kuala Lumpur, Malaysia, pp. 58-63, DOI: 10.1109/IE.2010.18, 2010.

Sawsan M. Mahmoud, Ahmad Lotfi, Caroline Langensiepen, Sherkat Nasser and Taha Osman, Echo State Network for Occupancy Prediction and Pattern Mining in Intelligent Environments, Fifth International Conference on Intelligent Environments, Barcelona, Spain, Vol. 2, pp. 474-481, DOI: 10.3233/978-1-60750-034-6-474 IE09, 2009.

Contents

List of Figures

List of Tables

Nomenclature

Greek Symbols

α Trend factor

λ Eigenvalue

π_i The probability of the first state $\pi_1 = P[Q_1 = S_i]$

τ Time delay length

Other Symbols

Dis Dissimilarity between two binary sets

D_x Input memory order

D_y Output memory order

B The probability of emission of symbol O from state S_i

S Hidden states

D Maximum adopted time-delay

r Residual matrix

Sim Similarity between two binary sets

T^2 $Hotelling's\ T^2$

t_{e_i} End-time of any event

t_{s_i} start-time of any event

V Eigenvector

Acronyms

ADL Activities of Daily Living

ADT Anomaly Detection Technique

AmI Ambient Intelligence

ANFIS Adaptive Neuro Fuzzy Inference System

ANN Artificial Neural Networks

AOFIS Adaptive On-line Fuzzy Inference System

BP Back Propagation

BPTT Back Propagation Through Time

DTW Discrete Time Warping

EM Expectation Maximization

ESN Echo State Network

FCM Fuzzy C-means

FSM Finite State Machine

GMM Gaussian Mixture Model

GP Genetic Programming (GP)

GSOM Growing Self-Organising Map

HC_HMM Hierarchical Context-HMM

HHMM Hierarchical HMM

HHSMM Hierarchical Hidden Semi-Markov Models

HMM Hidden Markov Model

iDorm intelligent Dormitory

IIE Intelligent Inhabited Environment

KNN K-Nearest Neighbour

LRN Layered Recurrent Neural Network

LSTM Long Short Term Memory

LWL Lazy Locally-Weighted Learning

MCD Minimum Covariance Determinant

MLP Multi-Layer Perceptron

MVE Minimum Volume Ellipsoid

NARX Non-linear Autoregressive netwoRk with eXogenous

OCSVM One Class Support Vector Machine

OPNN One-Pass Neural Network

PCA Principal Component Analysis

PC Principal Component

RMSE Root Mean Square Error

RNN Recurrent Neural Network

RSOM Recursive Self-Organising Map

RSSI Received Signal Strength Indicator

RTRL Real Time Recurrent Learning

$S-HSMM$ Switching Hidden Semi-Markov Model

SOM Self-Organising Map

SPE Square Prediction Error

SVM Support Vector Machine

FTDNN Focused Time Delay Neural Network

TDNN Time Delay Neural Network

WSN Wireless Sensor Network

Chapter 1

Introduction

The number of elderly people and people with physical disability who need peripheral help in their daily activities is rapidly increasing as revealed by the latest statistics on global population as reported in [1]. These statistics show an increase in the population group aged 65 or over. This will lead to a series of problems in caring for older people and people with disabilities. In addition, the European welfare model is not sufficient to satisfy the needs of the growing population, and increasing the number of care providers is not a realistic solution. It is recognized that using current technologies such as intelligent environments can help these people [2]. Intelligent environments can enhance the lifestyle of elderly people, keeping their privacy and letting them live in their own homes instead of care homes or hospitals for longer. As a result, costs of medical care for each person will be reduced.

Intelligent environments and specifically smart home environments [38] have become an important research topic in recent years. Smart home environments require systems able to detect, track and recognize people in their spaces. The following aspects may be identified by a smart home environment [86]:

- The number of people who occupy the environment,
- The occupants′ identification,
- The occupants′ physical activity, and
- The occupants′ localization in an area.

An important factor in designing a smart home for the elderly is that the technology should not interfere with normal activities. Thus, all devices should operate autonomously. To run autonomously without human interference, items and objects inside a house can be supplied with sensors to collect information on their usage. Examples of such sensors are house electrical devices such as cooker and fridge, domestic objects such as taps, bed and sofa, and temperature conditioning devices such as air conditioning and radiator. These items can be monitored or activated remotely. Some potential advantages of this technology may be [35,37]:

- Raising safety concerns, e.g. by monitoring behaviour patterns or current activities and providing help whenever a possible abnormal status is recognized,
- Comfort, e.g. by changing the temperature automatically, and
- Economy, e.g. by controlling the use of lights.

There are many smart home projects such as MavHome [38], iDorm [45], etc. addressing these issues.

The aim of this research is to investigate efficient mining of useful information from a sensor network forming an Ambient Intelligence (AmI) environment. In this thesis, we investigate methods for supporting independent living of the elderly (and specifically patients who are suffering from dementia) by means of equipping their home with a simple sensor network to monitor their behaviour and identify their Activities of Daily Living (ADL). Only low cost and readily available sensors are used which could be installed by the user themselves or their informal carers. These sensors are reliable and cheap [181] so they can be deployed in large quantities. Therefore, developing a technological solution easily retro-fitted in existing homes would definitely assist the elderly in gaining independence without altering their lifestyle or losing their personal dignity.

The rest of this chapter is structured as follows: in the next section, an overview of this research is presented. In Section 1.2, the aims of this thesis and the proposed objectives are presented. Section 1.3 introduces the major contribution of the thesis. Finally, the remaining chapters of this thesis are outlined in Section 1.4.

1.1 Overview of the Research

In an Intelligent Inhabited Environment (IIE), an individual user model can be learned from the sensory data which eventually represents the behavioural model of the user. Most work in modelling the behaviour of an occupant living in an intelligent environment and activity recognition has focused on using statistical methods such as Bayesian networks [11, 167] and Hidden Markov Models (HMMs) [80, 126, 144]. These statistical methods are employed to find the relationship between the data extracted from sensors and eventually identify the behaviour of an occupant. However, these methods experience difficulties in problems involving large low-level sensory data sets [160]. Also, the outputs from these methods are of significant network complexity [109]. The challenge for the research we face is to understand human behaviour from low level sensory data. We also face the challenge of interpretation of large amounts of sensor data. This could be achieved using common-sense knowledge (heuristic) or using computational intelligence integrated with sensory data. The current research primarily addresses elderly monitoring and well-being assurance in an IIE. For example by collecting movement activities only, we should be able to help an elderly person to live independently and raise an alarm in case of an emergency [36, 104].

The main research question addressed in this thesis is to investigate the use of sensor technology to analyse occupant behaviour. In particular, this study trying to answer the following questions:

- Can we extract behavioural patterns of a person from his/her ADL by analysing the time series data generated using occupancy sensors?
- Can we process the time series representing sensory data and predict the next step in the series in order to extract important daily patterns from them?
- Can we identify trends within occupancy sensory data?
- Can we identify unexpected patterns and anomalies within the data that is collected from sensors in an intelligent inhabitant environment?
- Can we validate and test these on data collected from real environments?

To answer the above questions, the aims and objectives of this research need to be expanded.

1.2 Aims and Objectives

The advances in sensor technology and availability of sensors have made it possible to easily measure various properties and activities of inhabitants in an IIE. However, obtaining meaningful knowledge from large amount of information gathered from a sensor network is not a straightforward task. Due to the complexity of inhabitant's behaviour, extracting meaningful information and ultimately predicting the values representing future activities of an occupant are research challenges [2, 64].

The aim of the work described in this thesis is to investigate effective analysis of the data collected from occupancy sensors in an IIE. This research tries to find an acceptable solution to monitor elderly people living independently in their own home. Most elderly patients would prefer to use a non-intrusive sensor technology [25]. These sensors would not affect their normal ADLs while other sensors such as wearable sensors need cooperation from the user to work properly and sometimes the user might forgot wear them [83]. This form of non-intrusive monitoring would not affect the normal daily activities of the system user. However, it will provide an early warning to the carer when an abnormal behaviour is monitored or expected to happen in near future.

In order to accomplish the aim of this research, the following objectives are identified:

- Use a sensor network system to extract daily behavioural patterns of the occupant in an IIE.

- Investigate different ways of representing and visualizing large sensor data sets in a format suitable for predicting and identifying the behavioural patterns. Most passive sensors produce long sequences of data as binary time series, indicating presence or absence of the occupant in different areas.

- Investigate appropriate means of pattern mining and prediction within the IIE to extract the behavioural patterns.

- Compare the performance of different Artificial Neural Networks (ANN) prediction techniques to assess the most appropriate technique for data collected from an IIE.

- Investigate different detection techniques to identify important information regarding outliers and any abnormal behaviour.

1.3 Major Contributions of the Thesis

The main contributions of this thesis are:

- Identification and prediction of the movement patterns of an occupant living in an IIE using only occupancy sensors.

- Investigate data compression techniques to be able to process and visualise large binary data sets collected from binary sensors. For example start-time and stop-time approach is shown to be more suitable for modelling and prediction.

- Investigate and determine frequent and abnormal user behaviours in an IIE. The approach is based on visualising and clustering sensor data sets in a format suitable for classifying and identifying abnormalities.

- Identifying outliers or anomalies within the behavioural patterns of an occupant in an IIE. The identification is based on different techniques including distance measures to classify user outliers activities and the severity of the outliers as well.

- Predicting users occupancy pattern based on ANN techniques including ESN and NARX network. Extensive experimentation are performed to validate the results.

1.4 Thesis Outline

This thesis consists of eight chapters that are summarized as follows:

Chapter 2: Literature Review
This chapter gives a review of the relevant literature in the field of intelligent environments. The main areas that are covered are the human behaviour recognition, representation and abnormal detection using statistical methods and computational intelligence techniques. In particular, the literature focuses on using available technologies for modelling the behaviour of the elderly people to support them to live independently in their own homes.

Chapter 3: Prediction Models
This chapter provides an overview of some existing techniques which are used in time series prediction. The chapter begins by presenting the traditional techniques such as HMMs as time series predictor. Then, the chapter introduces different ANN techniques used in this thesis and discuss their benefits in time series prediction.

Chapter 4: Environments and Data Collection
This chapter describes the system to monitor the ADLs for the elderly people. Two different environments including real and simulated environments are also explained in detail to validate and test the results. Details of the collected signals are also discussed.

Chapter 5: Data Representation and Visualization
In this chapter, the occupancy sensor data are represented using different techniques. In addition, the data interpretation and visualization approaches are discussed. These approaches are implemented on the raw sensor data to understand the movement's behavioural patterns of an occupant.

Chapter 6: Abnormal Behaviour Pattern Identification
In this chapter, some methods that uses the binary sensory data to identify the

normal behaviour and distinguish any abnormalities and possible trend in the behavioural changes of an occupant in an IIE are presented. A user activities outlier or anomalies detection system is proposed. The chapter starts with an overview of anomaly behaviour detection followed by the proposed system to identify the anomalies and outliers. This chapter closes by applying the proposed system to the environments discussed on the previous chapter. The chapter concludes that the proposed outliers and abnormal behavioural identification system is able to find anomalies or outliers within the sensor data. The severity of the outliers is also detected.

Chapter 7: Abnormal Behaviour Pattern Prediction
In this chapter, the results of the predictive models that are presented in Chapter 3 are validated using binary occupancy sensor data. A comparison between these models are made to find the best model to predict the presence or the absence of an occupant in an environment.

Chapter 8: Conclusions and Future Works
This chapter provides the conclusions arise from this thesis and formulates some future research in monitoring the daily activations of the elderly and disabled people in their own homes.

Chapter 2

Literature Review

2.1 Introduction

Monitoring system with appropriate measurement and communication equipment are available to support independent living for the elderly and disabled people. In particular, sensors can be installed into their homes to continuously monitor them and help to identify any deterioration in their health in a non-intrusive way. In addition to helping those people with less physical abilities, smart homes can also help the occupant to live their social life in a normal way. Today, a number of smart home projects have been developed. They are able to provide users more comfort, security, joy and welfare. More specifically, they are capable of monitoring the elderly people with motor, visual, auditory or cognitive disabilities [28]. This chapter reviews existing research studies on monitoring of human behaviour in an IIE. The review presents the existing studies on human behaviour representation, recognition, and abnormal detection algorithms and techniques. In addition, the identification and tracking of people who are living in smart homes are reviewed.

This chapter is structured as follows: in Section 2.2, the projects and technologies to support people at the early stages of dementia are reviewed. Some literature on human behaviour representation, recognition and abnormality detection are reviewed in Sections 2.3, 2.4 and 2.5 respectively. Conclusions are drawn in Section 2.6.

2.2 Intelligent Technology for Dementia Support

The European Union considers dementia to be one of the most important causes of disability in the elderly. Its figures show that between 1% to 2% of people aged $65-69$ suffer dementia, but this proportion more than doubles for people in the age band $70-74$, and studies across a number of countries show that its prevalence "increases almost exponentially with age" [2]. The socio-economic costs of dementia are large and increasing, and an international study by Anders Wimo of the Karolinska Institute suggest that 72.5 billion euros per annum accross the Europe is the cost of the informal care provided by family and other carers to dementia sufferers [64]. A further study by Wimo showed that carers have to spend many hours per day assisting dementia sufferers [182], and any technology that would reduce this would help to ease the costs - both financial and emotional.

A comprehensive survey published in [122] reports on state of the art technologies to support people at the early stages of dementia during the night. Extensive research has been reported on smart homes with a variety of applications including monitoring systems for elderly independent living, accident and fall detection [133, 192].

Research and development have focused mainly on the utilisation of different low-key technological devices which are readily available [23, 26, 143]. Most patients would prefer to use a non-intrusive technology to help them with their day-to-day activities. For example usage of surveillance cameras for patient monitoring is not welcomed and in most cases it is ruled out completely [103]. The major players in patient monitoring systems rely heavily on the use of monitored call centres rather than carers, with standard telephone lines for logging data and require significant installation. Some companies have realised the importance to patient care of an individualised system whereby the carers, relatives and others who know the dementia sufferers can monitor them; only intervening when the information and their personal knowledge indicates that the situation has changed significantly [99]. Research in independent living is not limited to dementia sufferers. Many published works address the issue of independent living in a broader sense [10, 24, 68, 75]. Smart homes can help to identify and model progression of dementia of the Al-Zheimer's type by evaluating performance in the execution of

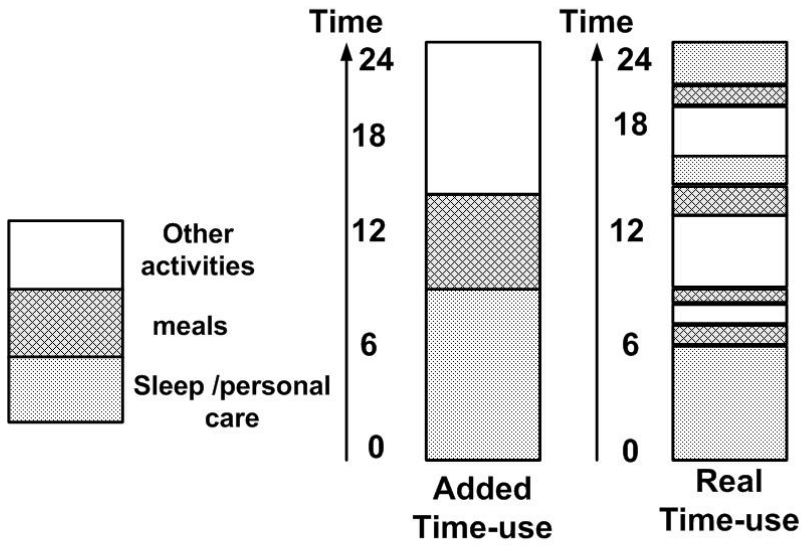


Figure 2.1: A time-use representation for 24-hours [177].

ADL [153].

Smart home can monitor and collect the activities information of the user by means of sensors, or communicate and control the environment. The former approach is widely used for monitoring [65], anomalous behaviour detection [126], behaviour diagnosis and prediction of activities in an ambient intelligence environment [7, 36, 166]. The latter approach is used to intervene and interact with the user as a means of preventing accidents and reminding the user. In the following sections, the literature on human behaviour representation, recognition and anomalies detection are presented.

2.3 Human Behaviour Representation

Different techniques are used for activity representation and interpretation which are extracted from an IIE. For example, the authors in [177] have incorporated a sequential pattern identification method to represent the user's movements during the day. Using this approach, the activities which are performed by an individual are represented using a single continuous vertical trajectory (time-use). A sample of time-use representation for the activities during one day is shown in Figure 2.1. Group of activities are added together to represent the duration time spent for

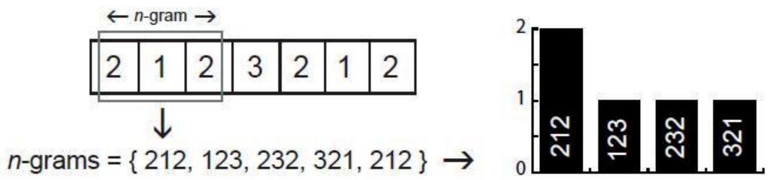


Figure 2.2: An example of transformation an activity sequence to histogram of event n-gram where n is equal to 3 [69].

each activity.

Authors in [69] have proposed an approach for activity representation in which the activities are treated as histograms of their event n-grams. An n-gram is adjacent activities of length n. Figure 2.2, shows an example of transformation an activity sequence to histogram of event 3-grams. In this kind of representation, the start and the end of an activity is assumed to be known and this activity is finished before another is started. One of the drawback of this transformation is that, it is necessary to choose a proper value of n that are able to represent the event dependence in an environment. As the value of n is increased, the order of information of events are more accurately captured. However, increasing the value of n affects the dimensionality of the histogram.

In [51] a set of attributes associated with sensors values including start-time, duration, weekend or weekday and activity level are identified. Authors in [12] proposed a method based on binary tree called Routine tree to represent the activities. This method associates the time periods to the most frequent patterns of an activity. By mining activity data, the routine tree is built to make it more compact. The data are collected from real sensors and the proposed method demonstrated high accuracy compression ratio. The tree routine can be combined for several days in order to identify the behavioural patterns of the user's routine.

Symbolic Aggregate approXimation (SAX) is a method which is used in [186] to do the delineation of the time series. In fact, the SAX representation takes the Point Aggregate Approximation (PAA) as an input and discretize it into a small alphabet of symbols. The discretization is achieved by imagining a series of breakpoints running parallel to the x-axis and labelling each region between the breakpoints with a discrete label. Any PAA value that falls within that region

Figure 2.3: A combined signal representation for four time series data extracted from binary sensors [7].

can then be mapped to the appropriate discrete value. Experimental results on several real data sets are also presented.

In [79], the movement data sets for a period of one week are sampled because of its large amounts of information. Sampling period is kept on the maximum value of the sampling period to evade any loss of information in the data. The sampling is done to compress the long series of the movement data and to build both the short-term and the long-term behaviour models of an elderly. For short-term model, the sampling period is only five seconds and two hours for the long-term model. After sampling, using a sliding window, a segment of the 10 time sequences is taken as one data point to model the user's behaviours. So, the next data point is the segment of one sampling period to the right.

Recently in [7], a combined signal is generated to represent a non-stationary time series assuming different levels for each activity. Figure 2.3 shows a combined signal representation for four time series data extracted from binary sensors. Each level of the combined signal represents one of the sensors. Then the signal is used to predict the future values of the time series data, which could be interpreted as prediction of the activities of an occupant in the environment.

2.4 Human Behaviour Recognition

Activity recognition is used to model the human behaviour in an intelligent environment and it becomes an essential element since it can be applied to many real applications. For example, it can be used to automate the health monitoring of human behaviour such as elderly people. The main challenges that may be faced in the nature of human activities are listed below [50]:

- Recognizing parallel activities: it means those activities that are done by people at the same time such as watching TV and talking with friends at the same time. Recognizing such activities requires using different techniques other than sequential activities recognition approaches.

- Recognizing overlapped activities: it means those activities that are overlapped with other activities in real live. For example, when people doing something in the kitchen such as cooking and the phone rings, the people stop cooking for a short period of time and after finish talking to their friend, they return back to their cooking and keep cooking.

- Vagueness in activities interpretation: it means that the similar activities may be interpreted in different ways depending on the current situation. For instance, the refrigerator door open activities may be understood either as a part of ”cooking” or ”cleaning”.

- Multiple occupants: it means that the environment is occupied by more than one person. In this case, those activities that are done in parallel by the occupants should be recognized, even though those activities are mutually performed by the occupants in a group.

There has been an increasing amount of literature on human behaviour recognition recently. In these literatures, different methods and algorithms are proposed for activity recognition. The key research challenge is to find a method or an algorithm that can efficiently recognize and model the human activities behaviour. A recent review on the activity recognition algorithms and techniques is presented in [30].

In this thesis, we classify these methods into two main categories, which are statistical and computational intelligence techniques presented in Section 2.4.1 and 2.4.2 respectively. Other techniques, that are used to recognize and model human behaviour, are reviewed and presented in Section 2.4.3.

2.4.1 Statistical Techniques

Most of the research, which has been carried out to deal with behaviour activity recognition and prediction, is done using statistical techniques. These techniques are used to find the dependence and correlations between the temporal data generated from sensors and ultimately identify the behaviour of an occupant. Probabilistic models become good techniques to identify human behaviour as they are capable of representing random variables, dependencies and temporal variation within data [14]. Several probability-based algorithms have been used to build activity models. The Hidden Markov Model (HMM) and the conditional random field are among the most popular modelling techniques [50]. Probabilistic models could be temporal such as Bayesian belief networks [11, 167] and HMMs [47, 59, 80, 126, 144]. In the following section, some commonly used statistical techniques for activity recognition are reviewed.

2.4.1.1 Bayesian Belief Network

Bayesian belief network is a statistical method which provides a more general framework to model human behaviour. These methods are used as a tool to process uncertain and incomplete data. For example, Naive Bayesian classifiers are used in [167] to classify and detect activities using tape-on sensor system. The authors used two types of activity recognition classifiers: multi-class naive classifier and multiple binary naive classifiers. The first classifier represents all the activities that are needed to recognize while the second one each classifier represents an activity to recognize. The major problem of Bayesian belief networks is the inflexibility of exact probabilistic inference [14, 109, 138, 191].

2.4.1.2 Hidden Markov Model

Hidden Markov Model (HMM) is one of the statistical models where a system uses a Markov process with unknown parameters. It consists of a number of hidden states and observations and is used to model human behaviour. Hidden Markov model is widely used to identify the activities of a user from sensor data. The hidden states represent the activities and the sensor data represent the observable output. For example, Hierarchical Hidden Semi-Markov Models

(HHSMMs) [102] are used to identify the daily activities of the occupants in an assisted living community.

The main issues when using basic HMM is the difficulties experienced in processing large low-level sensory data (i.e. temporal data from different time scales). Also, for each individual activity, the sequence of sensor event cannot be separated using these models [109, 160]. Moreover, using HMM in time series (where sensor data is represented as times series format) predictions require a large number of time series runs from HMM as the length of time series is increased [14, 47].

2.4.1.3 Finite State Machine

Recently, Floeck et al [54] use a class of Finite State Machine (FSM) called Mealy FSMs to locate the position of a person in a flat. The FSM is applied on datasets collected from sensor telegrams received for a period of time. Using Mealy FSM, important information are extracted from sensors. They are general activity/inactivity telegrams from different sensors, information regarding the occupancy of the environment and how the activity/inactivity patterns are changed after an emergency. The experiments are conducted on 30 occupants from two real-world projects.

2.4.1.4 Other Probabilistic Models

Other probabilistic models not mentioned above are used to recognize the human daily activities. For example, Expectation Maximization (EM) algorithm is utilized to minimize the uncertainty within the collected data. In particular, a version of EM algorithm called Monte Carlo EM is used to learn the parameters on-line. It is used to assess whether a system of basic movements sensors could distinguish between different behavioural patterns. Additionally, in [180], an individual behaviour is assigned through the unique transition probabilities between rooms and the activities. The simple way to learn the parameters of these transition probabilities is when the home is occupied by a single person identified by a RFID sensor.

Wen et al. [139], build a system able to find the location of the inhabitants in a smart home. The authors used the data collected from floor sensor to track the inhabitant. These sensors contain a lot of blocks each of them has a load cell in order to collect the human body weight. These blocks do not disturb the inhabitants when they walk since the blocks are covered with wooden flooring to gain a flat surface. Collected data are analysed to find the position of the inhabitants and follow their movements. In addition, the system can provide a history of the inhabitants' movements and predict their movements. The system is able to know where the inhabitant walks on the sensory floor in the smart home. However, one of the limitations within this system is that it can not distinguish different occupants if they have nearly same weight reading.

A mixture model structure is applied in [16] to build a probabilistic system of performance and examined on data from an intelligent environment system. Results are correlated with the inhabitant record to get a validation to the patterns. The influence of behaviour measurements during the working hours and off-days is independently tested.

In [67], a correlation between the night and day activities is achieved in the preliminary experiments. Passive infra-red sensors are used to detect the movement's activities of an elderly person in a hospital suit. The health of an inhabitant at early stage is predicted and the trends are identified. Also, in [159], correlated patterns are used to represent different activities. Correlated patterns are those patterns where higher occurrence exists within correlated activities. Activity recognition using correlated patterns is more accurate than frequent patterns in high dimensionality and large volume of data. However, simulation results are demonstrated only on simulated data generated from a testbed.

2.4.2 Computational Intelligence Techniques

As an alternative to the statistical methods, computational intelligence techniques are widely used to recognise the ADL. The following sections summarise some of these techniques.

2.4.2.1 Neural Networks

Artificial Neural Networks (ANNs) are used to deal with behavioural patterns collected from sensors networks in IIEs. Different combinations of ANNs are used in learning the daily routine activities of the occupant in an intelligent environment. For example, an approach named One-Pass Neural Network (OPNN)is used in [109] to detect the user's activities. The authors in that work performed that by adding a layer to the design of the network where the outputs of neural network are inputs to this layer. This layer consists of several cells including: static cell (expert knowledge), dynamic cell (temporal order) and decision making cell. The layer helps in differentiating between normal and abnormal behaviours based on the frequencies of ADLs.

Multi-Layer Perceptron (MLP) neural networks [43, 173] are used to identify the movements data collected from a WSN. For instance, the authors in [36], have applied different algorithms to recognize the age categories of data representing walking pattern and to identify the change in volunteers behaviour change. These algorithms are: MLP, decision tree, support vector classifier, Naive Bayes and Bayesnet. MLP gives the highest accuracy in classifying the categories, although, the size of training and testing data sets is small.

In [150], a prototype control system called Adaptive Control of Home Environment (ACHE) is developed. It is used to monitor an environment to understand the behaviour patterns in the environment and to predict the actions taken by occupants. ACHE is implemented using a feed-forward neural network trained with Back Propagation (BP) algorithm. The feed-forward neural network with BP is also implemented in [128] to control the basic occupant's living conditions such as air, heating, lighting, ventilation, and water heating. Authors in [194] have proposed a special kind of Self-Organising Map (SOM) in clustering the human daily activities. The proposed self-adaptive neural network is called a Growing Self-Organising Map (GSOM). Using GSOM, significant activity patterns in the data along with unusual data and abnormal behaviour can be revealed and detected. One major drawback of this approach is that the optimal learning parameters should be known in a priori such as an initial learning rate and the initial neighbourhood size.

For time series data modelling and prediction, much research is done using ANNs. Most of this research is carried out using feed-forward neural networks [55, 193]. In most cases, these networks connect input patterns to the output patterns without considering any feedback connections. Therefore, they do not take into account the temporal dependencies between the data. For example, in [7], different predictive techniques are used and compared to predict the occupancy behaviour patterns in an IIE. The occupancy data are collected from a WSN of door entry and motion sensors. The sensory data is combined to construct the binary time series to be inputs to the predictive techniques.

Recurrent neural networks are proven to be useful tools to solve the difficulties of the temporal relationships of inputs between observations at different time steps, by maintaining internal states that have memory. RNNs are computationally more powerful than feed forward networks and valuable approximation result have been obtained for prediction problems [72,162]. There are a few works attempted to use temporal neural network algorithms to detect, recognize and classify human activities in intelligent environments. For example, the authors in [84, 85] developed a temporal neural-network based agent, which can work with real-time data from unobtrusive low-level sensors and actuators, to identify human behaviour according to the temporal order of their activities.

2.4.2.2 Data Mining Techniques

Jakkula in [89], uses and compares different data mining techniques to classify and predict the data collected from an occupant living in an apartment. These techniques are: K-Nearest Neighbour , Support Vector Machine, MLP and Lazy Locally-Weighted Learning(LWL). They are also used as classifiers to predict the abnormal daily behaviour. However, some of their experiments need to be implemented on larger data sets to improve the prediction accuracy.

Data mining techniques such as association rules and Allen's temporal relations is investigated in [136] to identify everyday internal movement activities. An association rule discovers relationships between large sets of data elements. A simple activity identification based on C4.5 classification algorithm is presented and compared with the method of association rules. However, for a particular activity, the association rules method encountered a low rate of efficiency.

2.4.2.3 Support Vector Machine

Supervised learning techniques such as SVM are used in [100] to predict the occupant behaviour. The process enables predicting house holder's activities for frequent daily activities in the house such as grooming, eating, sleeping, having a breakfast, etc. The activity is identified for assessment initially. Then, SVM classifier is trained, using the datasets collected from sensors where the users perform their activities, by learning the user's habit. Although, the results are limited to the activities that are carried out at early morning only.

2.4.2.4 Fuzzy System

Fuzzy system can efficiently model the vague or uncertain data in sensor networks [7, 66, 118, 123, 124]. A system for ADLs recognition system is proposed in [123] using fuzzy logic. Fuzzy set theory is used to monitor the ADLs of an occupant to offer him/her a safe, comfort and an appropriate environment. However, in that research the fuzzy logic has been applied only on the simulated data.

In [7], fuzzy predictor model is used to build the prediction model and then the results are compared with the traditional time series prediction models such as ARMA, adaptive network-based fuzzy inference system and transductive neuro-fuzzy inference model with weighted normalization. One of the limitations of this work, the techniques does not apply for a complex and noisy data over a long period of time.

In [45], a fuzzy learning and adaptation approach for agents, called an Adaptive On-line Fuzzy Inference System (AOFIS) is proposed for ubiquitous computing environments. This approach consists of five phases including: monitoring the users' behaviour, capturing the actual data associated with their activities, computing the fuzzy membership functions from the input/output data, generating the fuzzy rules from the data and the agent control the learning and adaptation process. Thus, the intelligent agent has learnt, predicted and adapted to the needs of the user. To validate the proposed AOFIS approach, it is compared with other computational techniques such MLP neural networks, Genetic Programming (GP) and the Adaptive Neuro Fuzzy Inference System (ANFIS). The experiments are conducted on only five days in the intelligent Dormitory (iDorm) real environment and the results show that AOFIS results produce lower error predicting than both ANFIS and GP.

2.4.2.5 Machine Learning Techniques

Three algorithms were presented by [41] where machine learning techniques are employed to predict an inhabitant behaviour patterns, activities and common communication inside a home. The extracted data is utilised in automating decision making, frequent information and improving an inhabitant comfort, safety and efficiency. The movement of the inhabitant is predicted based on principles of information theory. Furthermore, another algorithm is set out on sequence matching in order to predict an inhabitant communications with the smart home, and also to identify significant patterns of the inhabitant activity.

The study in [93] concentrated on the task of whether a system including basic sensors could provide the basis for a methodology to predict the feature of the daily activity of the inhabitant. The system can detect and track the behavioural patterns on alarmed health issues. The process entails processing the sensor reading and combining these in order to develop a standardised prediction scheme using machine learning techniques. These schemes will help to assess the quality of life and the health of the occupants. On a long term base, the generation of prediction models on early stage of diseases and the generation of health variation patterns prediction could be achieved by using such real-time health monitoring systems. The KNN algorithm is exploited to predict the behaviour of the inhabitant for the next day.

2.4.3 Other Techniques Used in Behaviour Recognition

There are other techniques not mentioned above are used to recognize the human daily activities. For example, Hussain et al [81] have suggested an agent based architecture for knowledge discovery. Received Signal Strength Indicator (RSSI) technique is employed to scale the power of the signal at the receiver. The change in RSSI records is utilized for knowledge extraction. The following are examples of experiments carried-out:

- A Wireless Sensor Network (WSN) to find the sleeping patterns in a bedroom, as well as other physical activities carried-out by a person.

- A WSN to find whether the occupant's chair is occupied or not and to track the movements of the inhabitant in an environment,

In addition, in [8] a WSN is installed to collect data from various spaces in the AmI environment. The WSN consists of sensory agents of PIR sensors and door contact sensors. An RSSI detection system is incorporated to generate wireless localization agents together with tagging the occupant as a moving object to discriminate an inhabitant from other inhabitants or visitors.

2.5 Human Behaviour Abnormality Detection

As stated earlier, numerous studies have attempted to recognize the normal behaviour activities. Detecting the anomalies behaviour within an occupant's daily activities is another challenging task. Anomaly detection, also called the outlier detection, has been an important research area in many application domains. The general definition of anomaly detection is the problem of finding pattern in data that do not conform to expected behaviour. The non-conform patterns are often referred to as anomalies or outliers [188].

Many anomaly behaviours detection algorithms and techniques are proposed to solve problems in diverse domains including computer networks, medical, image processing, etc. In the following sections, some literature on anomaly detection techniques are presented and they are reviewed.

2.5.1 Statistical Techniques

Different statistical techniques are used to monitor the daily activities of an inhabitant in an IIE. A summary of the literature conducted results using statistical techniques in abnormality detection and outliers are presented below.

2.5.1.1 Gaussian Mixture Model

Gaussian Mixture Model (GMM) is used to learn the ADLs of an occupant in an intelligent environment in [51]. When new data arrives, an activity is detected where the activity's likelihood is estimated. Eventually, a decision is made that may be considered as a normal or abnormal behaviour. Rule based systems are used to make the decision to identify the ADLs in which each activity is associated with a set of attributes. Only normal behaviour are considered when the model is trained. Then any change from these behaviours is treated as abnormal and inform a caregiver for additional care and help.

2.5.1.2 Hidden Markov Model

In [126] a HMM is employed to model the occupant behaviour after using unsupervised classification techniques to group his/her daily routine activities. As a result, the model is able to detect the anomalies behaviour. HMM and K-means method are used in [127] to extract the bahavioual patterns in daily life. The collected sensor data is accumulated from a room environment. Then, for each segment of data, a behaviour description labelled is assigned which is computed using the HMM by likelihood of the segment. Based on the accumulated data, the probability density is composed using sequential discounting Laplace estimation and sequential discounting expectation and maximization algorithms. A score is calculated using logarithmic loss, which clarifies how much the new data behaviour is different from the stored one. Accordingly, if the score is high, anomalies are detected. When new data is available, the data is considered as abnormal if the behaviour changes dramatically. The main limitation of using this approach, however, is the differences of duration time of the behaviour do not take into account as an unexpected behaviour patterns. Only the frequency and the successive time of the behaviour are considered.

An extension of HMM is developed by Kang et al. [101] to model and learn the human behaviours and also to predict and identify the unexpected patterns inside a smart home. Using a tree like structure, the shared structured can be duplicated and presented. In particular, Hierarchical Hidden Markov Model (HHMM) is utilized to determine the abnormal behaviour activities considering the effect of the duration time of the activities and the relationships between them.

Chung et al. [32] propose a Hierarchical-Context HMM based human behaviour. The data is collected from a video stream in a nurse care centre. The HC-HMM includes three reasoning components: Spatial Context Reasoning (SCR) module, the Behaviour Reasoning (BR) module, and Temporal Context Reasoning (TCR) module. In HC-HMM approach, the behaviour is recognized by a sequence of spatial contexts which includes activities with temporal reasoning. One of the limitations with this study is that cameras have been used in monitoring the human behaviour. Another problem within this work is that their system fails to recognize daily behaviours with any starting and ending points.

Duong et al. [47] established a hierarchical model called Switching Hidden Semi-Markov Model (S-HSMM) a special form of the hierarchical model. This model consists of two layers, Markov sequence of switching variables in the top layer, and a sequence of concatenated HSMMs in the bottom layer in which its parameters are assigned by the top layer. Therefore, at the bottom layer, the dynamics and duration parameters of the hidden semi Markov model are switched from time to time and are not time invariant. This method of analysis has first to learn the normal activities by the model using training data. Then, the model can classify, segment, and detect anomaly activities.

Authors in [117] have clarified that explicit state duration HMM has advantage over implicit state duration HMM. The explicit state duration HMM can detect abnormal behaviour duration while implicit state duration fails in detecting abnormal deviation of the human behavioural patterns. Also, the variation of activity duration is important as the time order of the activity. Authors used a single camera to record data from 150 video sequences of normal activities in a kitchen.

HMM are also used in combination with other methods. For example, in [185] a combination of HMM and SVM is used to detect any abnormality within an elderly behaviour patterns. HMM is applied to train the normal ADLs and then SVM is used to classify the normal and abnormal behaviour.

2.5.1.3 Other Statistical Models

The basic statistical methods such as mean, standard deviation and z-score (also known as standard score) are used to identify outliers. For example, the authors in [90] investigated and built an outlier detection mechanism to improve energy management in a smart environment. The authors identified the outlier by finding the extremes using standard deviation. They also included a mechanism to rank the identified outliers to measure severity of the outlier. The z-score technique is used in [58] to detect outliers in ADL. The results are presented and tested using a volunteer in an apartment setting. If data is not normally distributed, mean and standard deviation are not good measures for detecting outliers.

An alternative approach is the box plot which is a graphical representation approach for examining data sets. For example, box plot is used in [40] to identify outliers which lie unusually far from the main body of the data. A box plot displays five important data summaries. They are: lowest value, lower quartile, median, upper quartile, and highest value. The advantages of the box plot are that it can display differences between populations without making assumptions about the underlying statistical distribution and the distance between the parts of the box indicates the degree of spread and skewness in the data set. However, it is argued that box plot is not an appropriate approach for every kind of data.

In a high dimensional vector, identifying outliers is a complex process. There are several methods which are used to find outliers in low dimensional space including Minimum Volume Ellipsoid (MVE) and Minimum Covariance Determinant (MCD) [22, 157, 163]. To process data with a high dimension, PCA has proven to be the preferred option. For example, in [163], an outlier detection system is introduced using PCA technique in combination with hierarchal clustering technique. They used cluster principal component analysis as a new distance-based method. The system is able to identify outliers in both single and multi-dimensional data.

2.5.2 Computational Intelligence Techniques

A summary of the literature using computational intelligence techniques in abnormality detection are presented below under different section headings.

2.5.2.1 Neural Networks

ANNs are used to distinguish between normal and abnormal human behaviour patterns from low-level sensors. For example, Illingworth et al. in [85] demonstrated that abnormalities can be related to the temporal order that the behavioural patterns activities are happened. In their study, a temporal ANN based embedded agent are proposed. Experiments are conducted on a real-time, data collected from low level sensors and actuators. Also, abnormalities in user's behavioural patterns are detected when new activities are appeared within these patterns. By adding rule nodes, ANN can adapt its hidden layer to hold new information whenever an example is not found to fit the existing structure.

2.5.2.2 Data Mining Techniques

Using data mining techniques such as model based clustering and association rules, the activities extracted from sensors are distinguished and the relationships between them are established in [168]. Additionally, FCM clustering technique is used in [79] to identify the boundaries of the normal behaviours. The data represents a movement activities for an elder person for a period of one week. The data characterizes the normal behaviour of the elder at his home. Then, any new data that is not belonged to any cluster are considered as abnormal. When an abnormal data is found, an alarm should be raised to the elder relatives, friends or caregiver.

Different clustering analysis techniques such as K-means and agglomerative hierarchical clustering are used in [115]. These techniques are used to find the similar movements patterns in a smart house. The large volume of data is interpreted using clock and scatter plots which show the differences between the movement's patterns.

The application of temporal relations is used in [92] to discover anomaly behaviours on the frequently-occurring events in an intelligent environment. The temporal relations are described the temporal order between events in the environment. In addition, by using these temporal relations the interval of time of each event in terms of start time and the end time values is identified, which are ultimately detecting anomaly behaviours. One major criticism of this work is that the temporal relations are applied on a sequence of small data sets to run experiments and validate the proposed system.

Matsuoka et al [121] developed a system able to detect anomalies within daily activities in a room occupied with different sensors. The data collected from such sensors are labelled and then accumulated extremely. A template is built from large amounts of the labelled data. When a new data is arrived, it is compared with template to compute a ratio. The ratio indicates a value of how much this data agree with the template. If the ratio is small then the new data is detected as anomaly.

2.5.2.3 Support Vector Machine

One Class Support Vector Machine (OCSVM) is developed in [91] for detecting abnormality within a low profile sensors in an intelligent home. The aim of that work was to let the inhabitants to live in their own home with no interference as possible. An algorithm offered by Weka was used as an incorporated tool to support vector classification and regression.

A system for abnormal activity detection are introduced in [189]. The system is based on data collected from wearable sensors and it consists of two stages. In the first stage a one-class SVM is developed for the normal activities. The SVM is able to sort out the normal activities which have a very high probability of being normal. In the second stage, the uncertain activities are treated since these activities need further detection. In this stage, a kernel non-linear regression analysis is used to separate the abnormal activity models from the normal activity models in an unsupervised method. One major drawback of that system is that several abnormal models may be built when abnormal activities turn out to be normal activities. This state may take place when the system monitors a user who repeat an activity continually after a certain period of time.

Support Vector Data Description (SVDD) algorithm is used in [156] to distinguish between normal and abnormal behavioural patterns. Using this algorithm, the abnormality is detected from a boundary around the target data is made by enclosing the target data within a minimum sphere. Several infra-red motion sensors are used to monitor elderly people living in intelligent environments to improve their personal healthcare system.

2.5.2.4 Fuzzy System

Fuzzy systems are widely used in smart homes to detect events in a WSN [120,123,124,163]. For example, fuzzy reasoning and three approaches including: statistics, association analysis and trend analysis, are proposed in [120]. The associations between activity patterns are carried out using an extended version of the Apriori algorithm. The normal behavioural patterns are learnt by observing frequent events and trends change in the ADLs. An alarm is raised when an abnormal event occurs. In that work, a sensor network system is installed in a home to identify a userś movements and the use of items such as furniture and household items. The actual sensor data is summarized and placed in a database to deal with as categorical data where, fuzzy membership functions are used instead of the actual time and duration of sensor reading.

In [13], a reliable identification of human activity using fuzzy logic is proposed. It identifies the state of a voxel person, a three dimensional representation of the human built in a real time. It is mainly used for modelling and monitoring of an elderly person falling from videos. Fuzzy logic models and monitors human activities in a way different from other human monitoring systems. It can produce interpretable information that can be used to understand, summarize the human activities and answer questions regarding changes in users behaviour and trends over times. In addition, fuzzy rules are extremely flexible as they can be modified, added or removed easily.

2.5.3 Other Techniques Used in Abnormality Detection

2.5.3.1 Use Cases

The application of use cases was studied by Tran et al. [169] to monitor an occupant for identifying illness, detect abnormal behaviours. Use case is a set of informal statements in natural language, which is used to describe a situation, and then generate outputs from the smart home. They are also used to recognize the context awareness and behaviours in an IIE.

2.5.3.2 Distance and Similarity Measures

Anomalous behaviours are also detected using distance and similarity functions. A considerable amount of literature has been published on (dis) similarity or distance measures [27, 49, 78, 146, 186]. These studies investigated the usage of these measures on a wide range of applications such as image retrieval, psychology, and biological taxonomy [27]. There has been little work so far to apply these measures in assisted living environment application and specifically human behavioural pattern identification. For example, in the aspects of sensor application, a similarity function [137] is used to train normal and abnormal sets of episodes using WSNs. A longest common subsequence algorithm is used to do the similarity. Then based on a threshold value, a decision is made to distinguish between normal and abnormal episodes of a sample of training set. In addition, weights are used with the similarity function which determined experimentally in order to get best results.

T^2 test of Mahalanobis distance in [188] is applied to identify the outliers and anomaly event detection in temporal data in an intelligent environment. The data set is generated using kernel smoothing method to give the results of one week template. These results can be used to present the daily activities in intelligent environments. The sensor data collected in first two months is used as training data set.

Rashidi et al. in [145] use Levenshtein (edit) distance to find the similarity between two patterns. The edit distance between two sequences is the number of edits (insertions, deletions, and substitutions) that are required to convert a sequence into another sequence. In that work, a new mining technique, called Discontinuous Varied-order Mining method (DVSM) is also proposed to find the frequent patterns that might be irregular and inconsistency in the ordering. To group the patterns into activity definition, a clustering technique is used in which the cluster centroids corresponds the activities that are identified and recognized. The main limitation of this work is that all the data points are clustered where only those that are part of an activity sequence is probably occurred often with a degree of constancy or recognisability.

In a recent study by Jakkula and Cook [90], an outlier detection system is proposed to preserve power in order to improve intelligent environments and make them efficient. They used KNN algorithms where Discrete Time Warping (DTW) is used for distance measures. It has been proved that KNN is well performed to differentiate between normal and abnormal data in intelligent environments. It is argued that, DTW will align the power value and it is better than the standard distance measures such as Euclidean, Manhattan, and Chebyshev measures.

2.6 Discussions

The knowledge gathered through this literature review suggests that it is possible to create the behavioural model of an occupant living in an environment equipped with sensors. Although the use of statistical methods are popular in extracting and predicting human behaviour, there are some problems associated with their utilization. For instance, HMMs have some problems in extracting multiple interacting either parallel or interference activities. In addition, due to its strict independence assumptions (on the observations), HMM is unable to grab the long-range or dependencies of the observations. Moreover; the observation sequences which are consistent with a specific activity may not be recognized using an HMM [50].

Recurrent neural networks have proven to be able to address the temporal relationships of input patterns since they incorporate feedback connections [72]. There are few studies which use temporal neural networks algorithms to detect, recognize, and classify the behavioural patterns of an occupant in a smart environment. These studies focus on using feed forward neural networks which does not include the time variant patterns.

The review presented here provided an overview of the anomaly detection and we discussed various anomaly detection techniques in the literature. Some limitations on using these techniques were also discussed in this chapter.

Some of the studies reviewed so far, have only been carried out in a small data size and implemented on data collected from simulated environment.

Chapter 3

Prediction Models

3.1 Introduction

In this chapter, prediction models are reviewed. Presented techniques are used later on in this thesis in Chapter 7 to predict the behavioural pattern of a user in an IIE based on sensory information. The prediction will help to provide us with information related to the user's health trend and to the carer to take an advance action. More details about the environment and signals are presented in the following chapters. Information collected from sensors within an IIE are considered as time series. In this chapter we mainly review prediction models which will apply to time series collected from sensor data. The investigated techniques are:

- Hidden Markov Model (HMM)
- Time Delay Neural Network (TDNN)
  - Focused Time Delay Neural Network (FTDNN)
  - Layered Recurrent Neural Network (LRN)
  - Non-linear Autoregressive netwoRk with eXogenous (NARX)
- Recurrent Neural Network (RNN)
  - Simple Recurrent Network

- Echo State Network (ESN)
- Long Short Term Memory (LSTM)
- Recursive Self-Organising Maps (RSOM)

In Section 3.2 an overview of time series is presented. The traditional time series prediction techniques are described briefly in Section 3.3. In Section 3.4, the theoretical basis of the HMM is reviewed followed by predictive ANN techniques. Conclusions of this chapter are drawn in Section 3.6.

3.2 Time Series

Time series data can be found in many everyday life applications including: scientific database with sensor data (e.g. weather, geological, environmental, astrophysics), financial application ... etc. [129]. Arranging a sequence of observations according to the time of their outcome is known as a time series model. Formally, a sequence of vectors at time t is denoted by:- $x(t)$, $t = 0, 1, 2, \ldots$.

For theoretical purposes, x can be presented as a continuous function of the time variable t. On the other hand, time viewing in terms of discrete time steps may lead to an instance of x at every end point of a fixed size time interval. Thus; it is called a time sequence or time series. The size of the time interval is usually problem dependant, i.e. it can be measured by milliseconds, minutes, hours, days and even years. Typically, these vectors capture changes to an object that is measured at equal time interval and are compound of any set of observable variables, such as:

- The temperature of air in a building,
- The price of a certain commodity at a given stock exchange,
- The amount of water consumed in a given community [46, 52].

The main goal in time series analysis is to develop some prediction techniques (models) for the time series $x(t)$. Then, using the developed model, identify the next time patterns from its past data and designing a control system based on the results of analysis. On the other words, the problem of time series prediction or

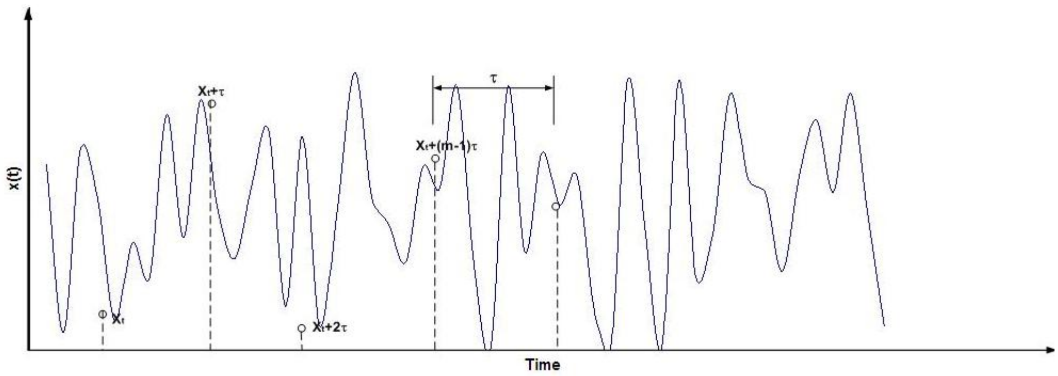


Figure 3.1: Continues time series data.

forecasting can be stated as follows: given a sequence $x(1)$, $x(2)$, $\ldots$, $x(t)$ up to time t, find the continuation $x(t+1)$, $x(t+2)$, $\ldots$. The most characteristic feature of time series is that they can be represented based on the original sequence by the values that may be randomly repeated several times without maintaining any definite periodicity [134, 178].

The values of time series data may be either **real** number (continuous) or **binary** number (discrete). Much research has been done in predicting continuous time series [46, 63, 88, 110, 170]. Continuous time series take real values, e.g., the temperature of a given room. Figure 3.1 illustrates a continuous time series data, where series $x(t)$, $x(t)+\tau$, $x(t)+2\tau$, $\ldots$ $x(t)+(m-1)\tau$ can be used as input variables to forecast the target variable, for all $t = 1, 2, \ldots n$ and a time delay length. Many techniques are applied to predict this kind of data. On the other hand, binary or (discrete) time series data has only value of 0 or 1 and the status of the data can be changed at any random time. Figure 3.2 shows a sample of discrete time series data. An example of a binary time series is the data collected from door entry sensor representing when it is open or close. The main challenge is how one can process this time series data and predict the next step in the series and extract important information from them.

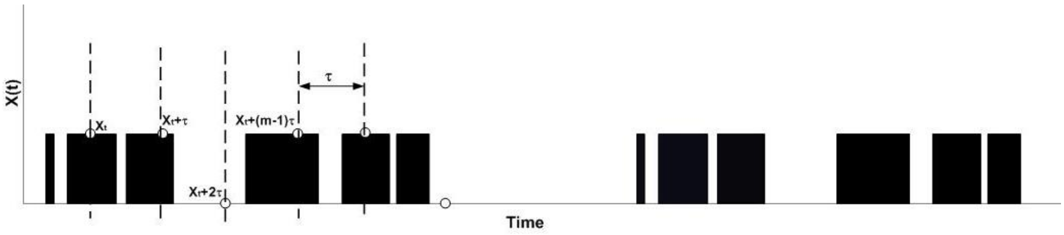


Figure 3.2: Discrete time series data.

3.3 Traditional Time Series Prediction Techniques

Many traditional techniques are used to process time series data. In most cases, the signal is assumed to be stationary and can be described by a set of linear equations. The most well known models are: AR (autoregressive), ARX (AR with eXternal input series), VAR (vector autoregression), ARMA (AutoRegressive Moving Average), ARMAX (ARMA with eXternal input series), ARIMA (AutoRegressive Integrated Moving Average), ARIMAX (ARIMA with eXternal input series), and ARFIMA (AutoRegressive Fractionally Integrated Moving Average). These linear time series models can be understood in great details since they have been studied for a long time. In addition they have tractability and ease of interpretation. Therefore, they are well developed and widely used. On the other hand, due to their linearity and simplicity, many complicated problem can not be implemented and many features of the underlying process are unsuccessfully captured. Consequently, this leads to unsatisfactory results in case of multi-step ahead prediction [77, 107].

Available alternative approaches are non-linear models, such as TAR (Threshold Autoregressive Model) and state-space models which are applied to model non-linear time series. A priori information, such as the type of the model and its complexities are required in these approaches. Owing to a large number of variables involved in most time series, as well as the high level of noise and limited amount of training data, this predefined information is not available for most time series and is difficult to obtain by estimation. For a particular problem these models work very well, for instance as reported in [77]; sunspots series prediction is well modelled using TAR.

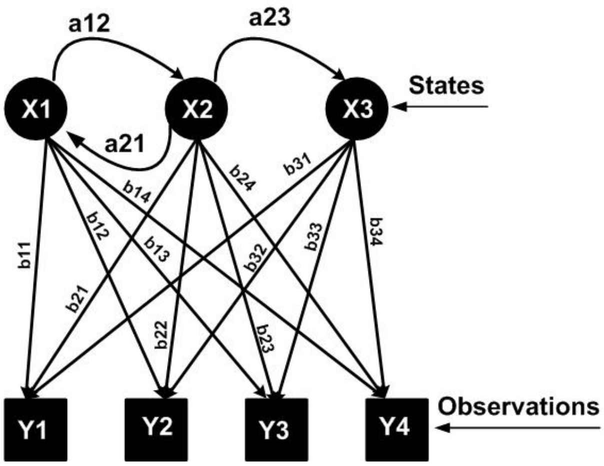


Figure 3.3: A graphical representation of Hidden Markov Model with three states.

3.4 Hidden Markov Models for Time Series Prediction

A discrete-time Hidden Markov Model(HMM) is defined by the following entities [18]:

- A set $S = S_1, S_2, \ldots, S_N$ of (hidden) states; where N is the number of states.
- A transition matrix $A = a_{ij}$, where $a_{ij} \geq 0$ represents the probability of going from state S_i to state S_j ;
- An emission matrix $B = b(O|S_i)$, where b represents the output probability and B indicates the probability of emission of symbol O from state S_i;
- An initial state probability distribution $\pi = \pi_i$, representing the probability of the first state $\pi_1 = P[Q_1 = S_i]$

A graphical representation of HMM is illustrated in Figure 3.3 where three hidden states and 4 observable variables are shown. Generally, the HMM can be represented as: $\lambda = P(A, B, \pi)$. These parameters should be chosen so that the model can well explain the observed data. HMM parameters are learnt using the Baum-Welch algorithm [183]. This algorithm is used to find out the parameters which maximize the probability of the observation variable of a model [18]. Also, a very common algorithm called forward-backward algorithm is used to find the maximum probability of a sequence O and given a model. In Appendix A, HMM algorithm is explained in more details [59].

For time series of multivariate values, HMM gives a probabilistic framework for modelling. Examples of HMM applications include; speech signal recognition, DNA sequence analysis, handwritten characters recognition, natural language domains etc. In most of these applications, HMMs are used either as a classifier or a predictor. HMM is a useful tool for time series prediction mainly for the following reasons [73]:

- when a new data is arrived, HMM is able to treat it robustly,
- due to the existence of established training algorithms HMM is computationally efficient to develop and evaluate, and
- it can predict the most frequent patterns efficiently.

Although, HMM can represent a time series; its induction algorithm has two essential disadvantages. Firstly, the number of states should be known in a priory which means the model is not fit to the data. Secondly, the algorithm can not converge to the global minimum [73].

3.5 Neural Network Techniques for Time Series Prediction

The challenges associated with time series modelling are lack of prior knowledge, high noise level, non-linearity and non stationary. A variety of artificial neural network ANN techniques have been proposed, investigated, and successfully applied to time series prediction. Power load forecasting, medical risk prediction, economic and financial forecasting, and chaotic time series prediction are all examples on application of time series prediction. In most cases, ANN prediction models demonstrate better performance than other traditional approaches [77].

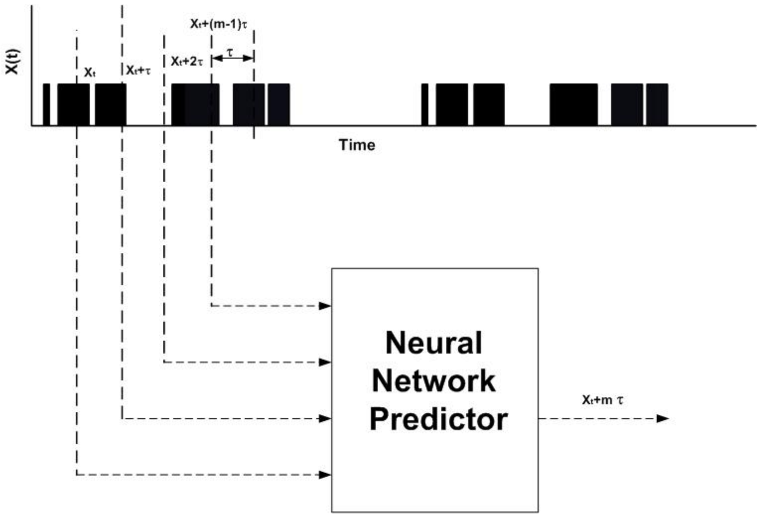


Figure 3.4: Neural networks for time series processing.

ANNs are data-driven, self-adaptive non-linear methods that do not require specific assumptions about the underlying model. Instead of fitting the data with a pre-specified model form, neural networks let the data itself serve as direct evidence to support the model's estimation of the underlying generation process. This non-parametric feature makes them quite flexible in modelling real-world phenomena where observations are generally available but the theoretical relationship is not known or testable. Accordingly, a large number of successful applications have verified the role of ANNs in time series modelling and forecasting. It also distinguishes neural network models from traditional linear models and other parametric non-linear approaches, which are often limited in scope when handling non-linear or non standard problems [193].

In this research, selected methods of the ANNs are used in prediction the future time step of discrete values of time series data. Figure 3.4 shows a diagram illustrating the inputs to a network as $x(t)$, $x(t)+\tau$, $x(t)+2\tau$, $\ldots x(t)+(m-1)\tau$ which is capable of predicting the next time step $x(t)+m\tau$.

The following sections will review the most common neural network techniques used in processing and prediction the time series data. These techniques include, Multilayer Feed-forward neural networks, Time Delay Neural Networks, and Recurrent Neural Network.

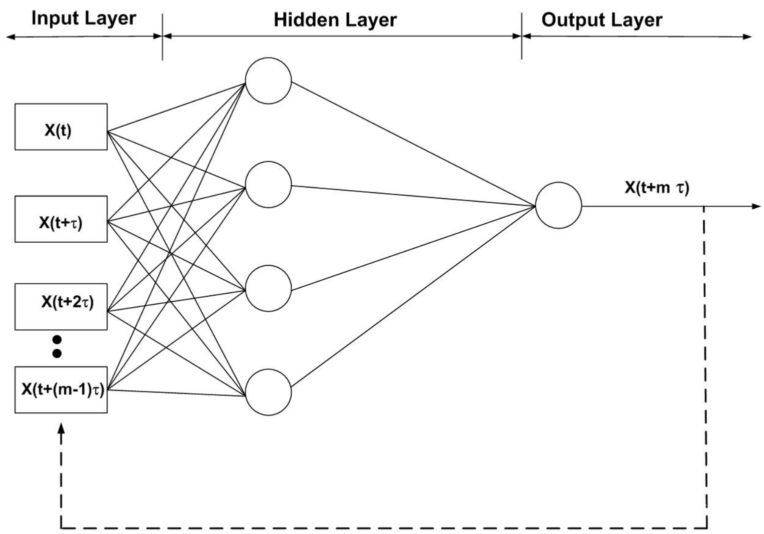


Figure 3.5: Time-lagged input variables as the input layer of neural network. The output of the model $x(t + m\tau)$ is one step ahead prediction.

3.5.1 Multilayer Feed-forward Neural Networks

One of the challenges of applying static ANN models to time series prediction is to incorporate the temporal relationship between observations at different time steps into the model. The simplest way to include temporal information into a multilayer feed-forward network is by using different time-lagged input variables. Figure 3.5 illustrates a multilayer feed-forward with $m - 1$ time-lagged input variables. Selecting proper time lags and an informative set of input variables is critical to the solution of any time series prediction problems. Since choosing suitable time lags is a difficult problem, another practical approach is to first select as many lagged input variables as possible, then apply Principal Component Analysis (PCA) to the input space and transform input variables into new variables in the principle component space, which usually has a much lower dimensionality than the original space. Transformed variables are then used to train neural networks [77].

Among many ANN techniques used in forecasting, the single multilayer feed-forward model or Multi Layer Perceptron (MLP) is known as the most popular one. It is proven theoretically that the MLP has a universal functional approximating capability and can approximate any non-linear function with arbitrary accuracy [193]. However, the algorithm yielded no unified guideline in choosing the appropriate model structure for practical applications. Thus, a trial-and-error approach or cross-validation experiment is often adopted to help find the "best" model. Typically a large number of ANN models are considered. The one with the best performance in the validation set is chosen as the winner, and the others are discarded [193].

3.5.2 Time Delay Neural Networks

In this section, another class of ANNs so called Time Delay Neural Networks (TDNN) is introduced. TDNNs rely mainly on special kind of memory known as **tap delay line** where the most recent inputs are buffered at different time steps. Such delay lines between hidden and output layers are necessary to supply the network with additional memory. In other words, by using delay lines the inputs arrive hidden layers at different points in time, so they stored long enough to influence subsequent inputs. A typical tap delay line is illustrated in Figure 3.6. The response of these ANNs in time t is based on the inputs in times $(t-1)$, $(t-2)$, $\ldots$, $(t-D)$. A mapping performed by the TDNN produces a $y(t)$ output at time t as:

$$y(t) = f(x(t),\ x(t-1),\ \ldots,\ x(t-D)) \tag{3.1}$$

where $x(t)$ is the input at time t and D is the maximum adopted time-delay. TDNN is well suited in the applications of speech recognition and time series prediction. In prediction, TDNN can deal successfully with the dynamic behaviour of the system and predict the next state [46, 119].

Although all the connections in the TDNN are feed-forward, which is similar to MLP, the inputs to any unit in the network have the output of the previous stage. The activation of the unit f at any time step is calculated as follows:

$$y_i^t = f(\sum_{j=1}^{i-1}\sum_{k=0}^{d} y_j^{t-k} \cdot w_{ijk}) \tag{3.2}$$

where y_i^t is the output of node i at time t and w_{ijk} is the weight to the node i from the output of node j at time $t-k$. TDNN are used successfully for prediction,

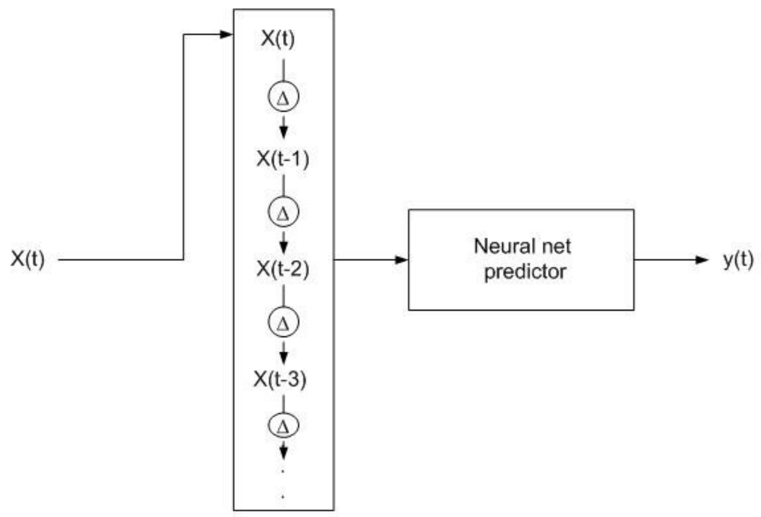


Figure 3.6: Tapped delay line memory model. Each delta operator introduces a one step time delay.

because they are able to capture the dynamics of a system and to foresee the output in the current time [33, 119]. TDNN is sometimes called Neural Network Finite Impulse Response (NNFIR).

Three different TDNNs are investigated to predict an approximate value for the future samples of the series. These include: Focused Time Delay Neural Networks(FTDNN), Layer Recurrent Network (LRN) and Non-linear Autoregressive netwoRk with eXogenous (NARX). Different sizes of tapped delay line have been attached in order to predict the next samples of the input.

3.5.2.1 Focused Time Delay Neural Networks

Focused Time Delay Network is a MLP with a tapped delay line (also called memory layer) as input layer. A typical diagram of focused time delay neural network is illustrates in Figure 3.7. This network belongs to a class of dynamic networks. Delay time line is used to store the historical samples of the inputs. The number of historical samples determine the size of the memory layer to express the features of the input in time. The memory is always at the input of multilayer feed-forward networks; hence the name focused comes.

Training in focused time delay network is much faster than other dynami-

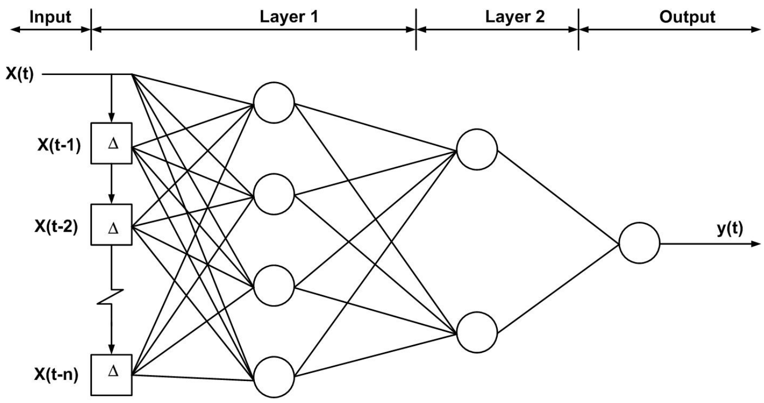


Figure 3.7: Focused time delay neural network with two layers.

cal network for two reasons: firstly, as mentioned before tapped delay appears only at input layer and secondly, the loop does not contain feedback connections or adjustable parameters. For that reasons, no dynamic back propagation are needed to compute the network gradient. It can still be trained with static back propagation [42, 141].

3.5.2.2 Layer Recurrent Networks

Another class of dynamic network is layer recurrent neural network which is originated from Elman network (it is explained in Section 3.5.3.1). Unlike focused time delay neural network, the time delay lines in this network are found in each layer except the last layer. A layer recurrent network which included two layers with feedback connections is illustrated in Figure 3.8. The key modifications to the Elman network are different number of layers and transfer functions are used, whereas, the original Elman network uses two layer with a sigmoid function for hidden layer and linear function for the output layer. Back propagation algorithm has been implemented to train the Elman network [42].

3.5.2.3 The NARX Network

Another type of dynamic network but with feedback connections is Non-linear Autoregressive netwoRk with eXogenous inputs(NARX). The NARX model is a discrete-time non-linear model which is demonstrated to be equivalent to Turing

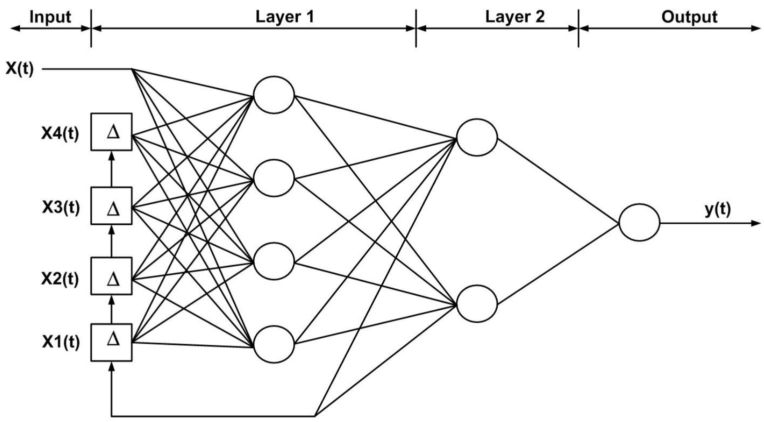


Figure 3.8: Layer recurrent network with two layers.

machine [98]. Figure 3.9 shows the typical architecture of NARX network [164]. In this model, multilayer perceptron network is used to approximate $y(t)$ expressed as:

$$\begin{aligned} y(t) &= f(x(t-1), x(t-2) \ldots x(t-D_x) \\ &\quad , y(t-1), y(t-2) \ldots y(t-D_y)) \end{aligned} \tag{3.3}$$

where $x(t)$ and $y(t)$ are respectively the input and the output of the model at time step t, while D_x and D_y are the input and the output memory orders with $D_x \geq 1$, $D_y \geq 1$ and $D_y \geq D_x$. The non-linear function of the input and output of the model is expressed as f. The predicted output $y(t)$ is regressed on the input value (exogenous) $x(t-1)$ and the output value $y(t-1)$ [98, 158]. In this case, since one of the inputs of NARX is the output of the network, this makes NARX network represent the dynamical characteristic of a system efficiently [96]. NARX network can also be implemented as TDNN when its output memory order takes a zero value. Accordingly, there is only a time delay line in the input layer of the MLP learning algorithm which is used to approximate the following function in TDNN [113]:

$$y(t) = f(x(t-1), x(t-2) \ldots x(t-Dx)) \tag{3.4}$$

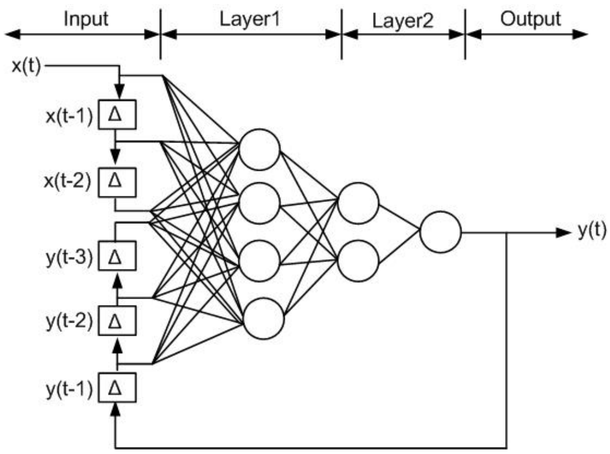


Figure 3.9: NARX Network with two input delays and three output delays.

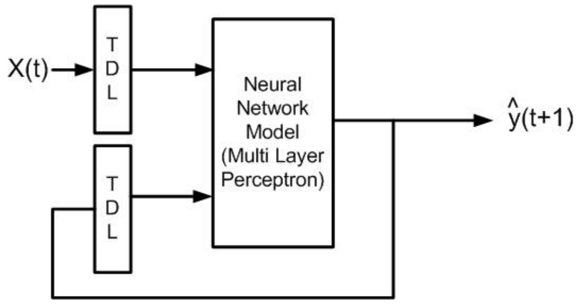
Generally, there are two modes in which NARX network is trained [96,184]. These modes are:

1. Parallel Mode: In this case, the estimated output of the network is returned to the input of the MLP as shown in the following equation:

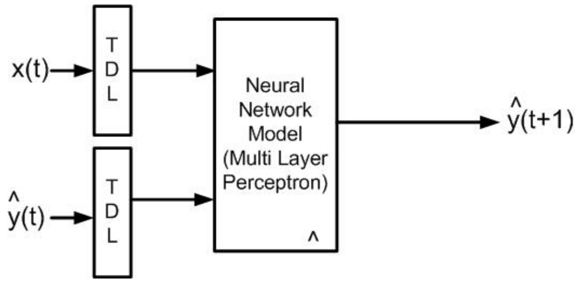
$$\begin{aligned} \hat{y}(t+1) &= \hat{f}[(y(t), x(t)] \\ &= \hat{f}[y(t), y(t-1), \dots y(t-D_y+1); \\ &\quad x(t), x(t-1), \dots x(t-D_x+1)] \end{aligned} \tag{3.5}$$

   where $\hat{f}$ is a non-linear function of the input and output of the model which is approximated by MLP. This mode is shown in Figure 3.10-a where TDL refers to the tapped delay line. This mode can provide a good estimation when one adds a regressive factor of the estimated value. As a result, the main dynamic features of the system are obtained.

2. Serial-Parallel Mode: In this case, instead of feeding back the estimated output of the network, the actual output is returned to the input of the



(a) Parallel network



(b) Serial-Parallel network

Figure 3.10: NARX network architecture (a) parallel network, (b) serial-parallel network.

neural network as shown in the following equation:

$$\begin{aligned} \hat{y}(t+1) &= \hat{f}[(\hat{y}(t), x(t)] \\ &= \hat{f}[\hat{y}(t), \hat{y}(t-1), \ldots \hat{y}(t-D_y+1); \\ &\quad x(t), x(t-1), \ldots x(t-D_x+1)] \end{aligned} \tag{3.6}$$

where $\hat{f}$ is a non-linear function of the input and output of the model. Figure 3.10-b illustrates the serial-parallel Mode of NARX network. This mode has an important characteristic in that it has strictly feed-forward architecture and a static back propagation learning algorithm can be used.

The embedded memory of NARX network gives a shorter path for gradient information in case the network is unfolded in time to back propagate the error signal. Having such characteristic, the gradient descent learning is better in NARX network in learning the long term dependencies. In gradient-based training algorithms, for n time steps in the past, the fraction of the gradient becomes zero as n increases. Vanishing gradient is a problem in other neural networks such as back propagation through time [98, 158].

3.5.3 Recurrent Neural Networks

Most neural networks algorithms for predicting chaotic time series are based on feed-forward neural networks, mentioned in earlier section. Defining a suitable number of hidden nodes is a difficult problem of these networks. Besides, being static networks, these networks have limitations to identify chaotic dynamical systems [72, 162].

Time series prediction problems can be solved using Recurrent Neural Networks (RNNs) because of the dynamic nature of these networks. Thus the temporal relationship of inputs between observations at different time steps is handled. These networks are capable of maintaining related historical information for predicting the future trend of the series [19, 77].

RNNs are proven to be effective in learning time-dependent signals that have short term structure [162]. For signals with long term dependencies, these networks are seemed to be less successful, as during training, the error gets “diluted” when passed back through the layers many times. However, RNNs are computationally more powerful than feed-forward networks and valuable approximation results were obtained for prediction problems [162]. Moreover, recently, a number of researchers have added connections with time delays to the RNNs which often allow gradient descent algorithms to find better solutions in these cases. The training of RNNs tends to be more difficult because of the feedback connections. Real time recurrent learning and back propagation through time are the two popular training algorithms of RNNs [72, 162].

RNN techniques are investigated later on and are explained in the following sections.

3.5.3.1 The Elman Network

In this section Elman recurrent neural network also known as a partial recurrent network is explained [48]. Figure 3.11 shows the structure of the network with two inputs and one output unit. Elman network has three layers: input, hidden

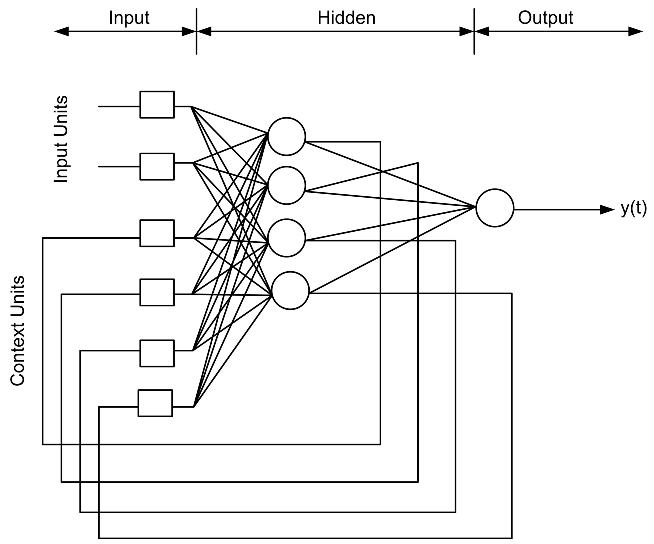


Figure 3.11: Elman network structure.

and output layers, with a number of "context units" in the input layer. Initially, the context units take zero values and then the output values of the hidden layer units at the previous time steps will be copied into these units. In this case, the network can perform time function mappings that are beyond the ability of the standard MLP. This characteristic allows Elman network to be suitable for time series prediction. Accordingly, the memory of the Elman network is constructed via a feedback. Using this feedback, temporal and spatial patterns can be learned, recognized and generated [57, 61, 174].

Elman network is also trained using a discrete-time RNNs training algorithm known as Back Propagation Through Time (BPTT). It is well known neural network training algorithm in many temporal classification problems. However; to achieve high accuracy in the learning process, many computational times are required [140, 172]. Figure 3.12 illustrates the scheme of BPTT.

BPTT training algorithm consists of two passes; forward and backward pass. In forward pass, the stacked network starting from the first copy till the end of the stack are updating in one training epoch. In Appendix B, the BPTT algorithm is explained in more detail [87]. At the beginning, the outputs of BPTT algorithm are computed for all time steps. Then, the gradient is calculated by starting at the last time step and working backward in time. The main disadvantages of the

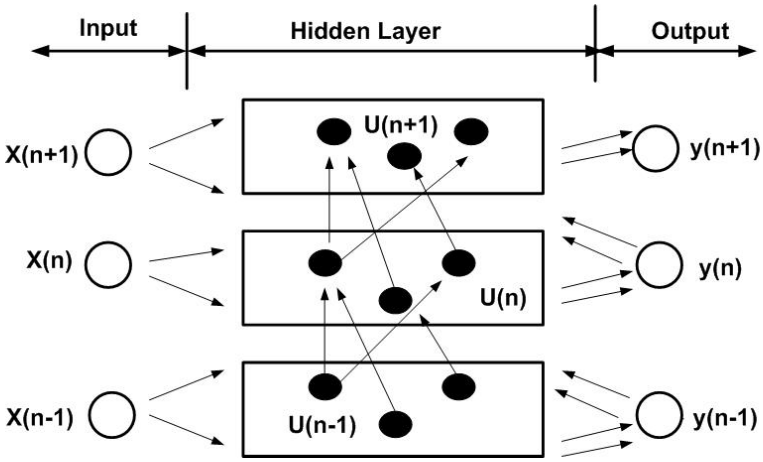


Figure 3.12: Scheme of the basic idea of Back Propagation Through Time.

BPTT algorithm is that it is not suitable for on-line applications because it works backward in time from the last time step. So this algorithm is more powerful in the gradient calculation [95].

Many applications have been effectively implemented by using back propagation through time such as: pattern recognition, dynamic modelling, sensitivity analysis, and the control of systems over time. Also, it can be used employed with neural networks, econometric models, fuzzy logic structures, fluid dynamics models and with many other systems [179]

3.5.3.2 Echo State Network

In this section, the recurrent neural network, Echo State Network (ESN) is described. It was developed recently by Jaeger [88]. The basic architecture of ESN is illustrated in Figure 3.13 which consists of three layers. These include input, hidden and output layer. The input layer is connected to the hidden layer. Only, the hidden layer are fully connected to the output layer. On the other hand, the output layer is backward connected to the hidden layer only. It is a discrete-time, continuous state where the activation function for all neurons is the sigmoid function [11].

An ESN consists of a reservoir of conventional processing elements, which are recurrently interconnected with untrained random weights, and a readout (output) layer, which is trained using linear regression methods. The key advantage

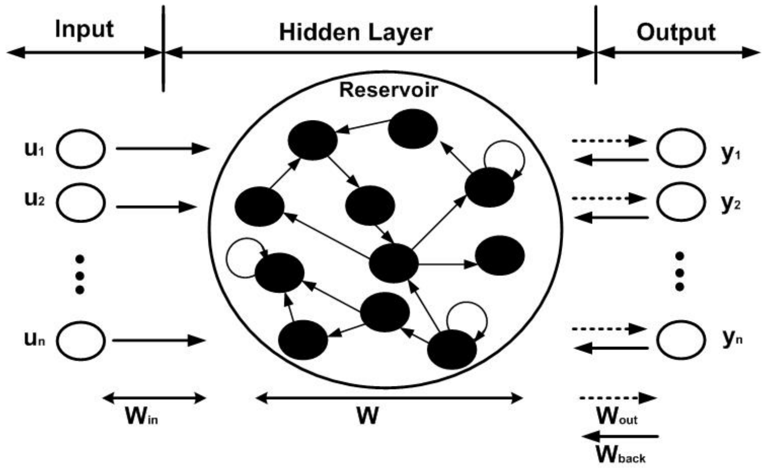


Figure 3.13: Structure of an echo state network approach. Only the output weights W_{out} are adapted, all other weights (input, reservoir and feedback) are chosen randomly.

of the ESN is its ability to model systems without the need to train the recurrent weights [161]. For training an ESN with an input u_n, a reservoir state $x(n)$ with M processing elements, and an output y_n, the equations are calculated as follows:

$$x(n+1) = tansig(w_x \times x(n) + w_{in} \times u(n) + v(n+1)) \tag{3.7}$$

and

$$y(n) = w_{out} \times x(n) \tag{3.8}$$

where $x(n)$ denotes the hidden layer or the internal state. $tansig$ denotes hyperbolic tangent sigmoid function which is applied element wise, $v(n+1)$ is an optional noise vector. w_x, w_{in} and w_{out} are respectively the internal connection weights of the reservoir, the input weights to the reservoir and the readout (output) weights from the reservoir [155].

The ESN approach differs from other methods in that a large RNN is used (in the order of 50 to 1000 neurons) and in that only the synaptic connections from the RNN to the output neurons are updated i.e. weights coming from the hidden layer (i.e. the reservoir) to the output layer are updated in order to achieve the learning task. As a result, large datasets are learnt in only a few minutes or even seconds [88]. Also, there are neurons in the reservoir connected in loops (see Figure 3.13), therefore the past states 'echo' in the reservoir.

The convergence of training in ESN is much faster than other RNN. This has made ESN an attractive model for a wide range of time series prediction, pattern generation, event detection and classification. The prediction is accomplished using a black box model, i.e. it only depends on past data since no further information is used [76, 132, 155].

3.5.3.3 Long Short Term Memory

In this section, a recurrent neural network known as Long Short Term Memory (LSTM) is briefly described [151]. One of the most important characteristic of this algorithm, which is different from traditional RNNs techniques, is its ability to solve vanishing gradient problem. Vanishing gradient problem means the influence of a given input on the hidden layer and therefore on the output of the network, either vanishes or blows up exponentially as it cycles around the recurrent connections. In most RNNs techniques, the errors flow backwards in time tends to either blow up or vanish. LSTM learning algorithm can enforce constant error flow through, hence neither exploding nor vanishing. LSTM network is an efficient algorithm with application regarding temporal processing tasks [70, 131].

LSTM network have three layers: input, hidden and output layer as shown in Figure 3.14. The major difference of LSTM network from other RNNs is its hidden (internal) units. The basic units of the hidden layer are memory blocks containing one or more memory cells and three adaptive and multiplicative gating units shared by the cells in the block. The memory cell has connection to its self with a weight of value one called "Constant Error Carousel" (CEC). The function of CEC is to provide short-term memory storage for extended time periods by recirculation activation and error signals indefinitely. All the inputs are connected to all of the cells and gates. The cells are connected to the outputs and the gates are connected to other cells and gates in the hidden layer. The input gate controls the flow of activation into the cell by passing the input to the memory cell through a squashing activation function of the input gate. On the other hand, the output

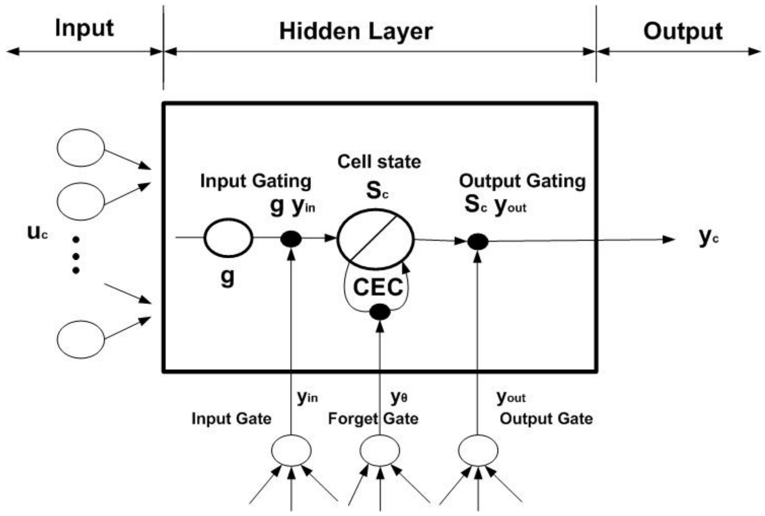


Figure 3.14: A schematic diagram of LSTM memory block with one cell and its gate units.

gate controls the flow of the activation from cell to the outputs. The memory cell's output is passed through another squashing function before being gated by the output gate activation [151].

Two passes are required to train LSTM: forward and backward pass. All units' activation are updated in forward pass, and then the errors signals for all weights are computed in backward pass. The pseudo code details for forward pass, backward pass, and weight updates LSTM training algorithm is included in Appendix C. Many real world sequence processing problems have been implemented using LSTM recurrent learning algorithm [60, 151].

3.5.3.4 Recursive Self-Organising Map

Recursive Self-Organising Maps (RSOM) is an extension of the classical prosperities of Self-Organising Maps (SOM) [175]. RSOM is created by adding feedback connections to SOM and keeping its original self-organization property. These feedback connections are used to represent time implicitly and self-referent. RSOM learns local representations of the temporal context associated with a time series [176].

The architecture of RSOM is shown in Figure 3.15. The inputs to RSOM network are the input and the time-delayed (context) copy of the activities which

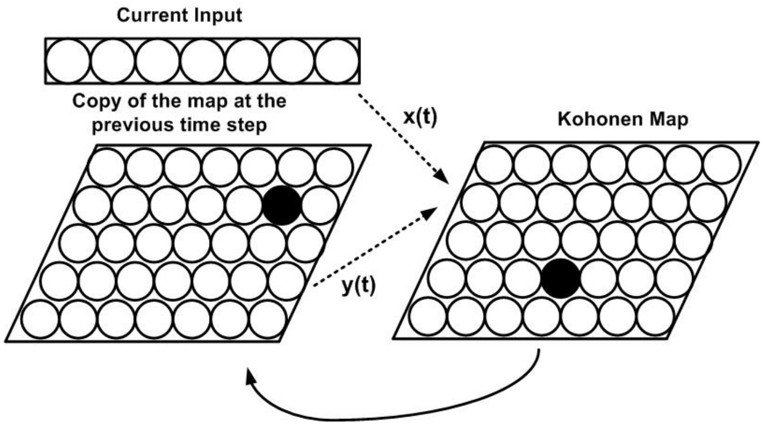


Figure 3.15: An architecture of recursive self-organising maps. Trainable connections are represented by the dotted line while the fixed connections are represented by the continuous line.

are the single input vector to SOM network topology. In this network, the map is able to learn both the input and the context. Based on the learned shorter sequences, the long sequences of inputs are learned iteratively.

In RSOM, each unit i of the map ($1 \leq i \leq N$) has a receptive field defined by two weight vectors, $w_i x$ and $w_i y$, that are compared to the input vector, $x(n)$, and to the vector of activities in the map at the previous time, $y(n-1)$, respectively. The activity, $y_i(n)$, of unit i at step n is:

$$y_i(n) = exp(-\alpha \|x(n) - w_i^x\|^2 - \beta \|y(n-1) - w_i^y\|^2) \tag{3.9}$$

where α and β are constant coefficients. The best matching unit is the unit that has the highest activity. If k is the index of the unit maximising $y_i(n)$, the learning rules used for the weights are:

$$\triangle w_i^x = \gamma(h_{i,k}(x(n) - w_i^x) \tag{3.10}$$

$$\triangle w_i^y = \delta(h_{i,k}(y(n-1) - w_i^y) \tag{3.11}$$

where γ and δ are learning rate, and the neighbourhood function $h_{i,k}$ is a Gaussian function of the Euclidean distance, $d(I,k)$, between units i and k on the map:

$$h_{ik} = exp(-d(i,k)^2/\sigma^2) \tag{3.12}$$

Here, σ controls the width of the Gaussian [176].

3.6 Discussions

This chapter gives an overview of different techniques as a solution for time series prediction. The predictive techniques includes statistical methods such as HMM and ANNs. We have seen that how the dynamic nature of RNN techniques help in handling the temporal relationships of inputs between observations at different time steps. In particular, it has been shown that these networks are capable of maintaining related historical information for predicting the future trend of the series [19, 77]

Chapter 7, will describe the implementation of these predictive techniques and compare between them. The techniques are evaluated using data sets collected from ADLs of occupants living in houses equipped with an appropriate sensors and door entry sensors. Those sensors are used to record the behaviour of the occupant, and allow the carer to observe any changes to patterns.

In the next chapter, a description of data collection and environments will be introduced. Data collection system is employed in this research in order to monitor ADLS of an elderly person.

Chapter 4

Environment and Data Collection

4.1 Introduction

In a smart home, it is important to know when the occupants carry out the most activities so that more assistance and support may be allocated to them. Also, being aware of the most frequent daily activities may also aid in determining any future irregular patterns within a daily routine such as spending a long time in the bedroom, relentless roaming around the house, or unusual absence for long periods and so on [127]. Therefore, it is vital to find a robust system able to monitor the daily activities of the people with less ability. Also, the adult children of frail elders living alone and at a distance could be sent reports or alerts daily/weekly in the form of e-mail or phone calls, and they could even be informed if any abnormality in the near future is predicted.

This chapter gives an overview of intelligent environments including sensor networks. The description of the procedure for data collection from a sensor network to monitor the daily activities of an elderly person is also presented. In addition, trend as an important component in activities of daily living is modelled and integrated within a single-occupant occupancy simulator, So that in the occupancy signal generated by the simulator, both seasonality and trend are included in occupant's movements. Different types of trend, in the occupancy signal, improve the occupancy modelling by enabling the model to incorporate long term differentiation in occupant's behaviour i.e. ageing, health, and other

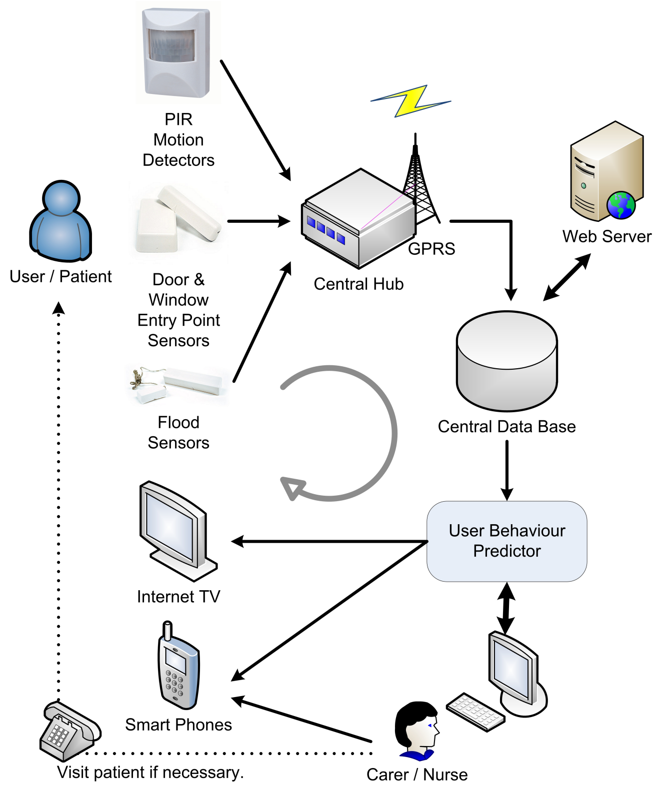


Figure 4.1: An overview of monitoring and interaction system architecture.

changes in his/her activities of daily living.

This chapter is organized as follows: in Section 4.2, an overview of ambient intelligence is presented. Section 4.3 introduces the sensor network in an intelligent environment and data handling. In Section 4.5, a description of the daily activities monitoring of an elderly person living in a smart home is presented. Sensor data collection is presented in Section 4.6 where two different environments are explained in detail. Some conclusions are drawn in Section 4.7.

4.2 Ambient Intelligence

An Ambient Intelligence (AmI) is a new information model in which people are monitored by a digital environment where their daily activities along with their needs are responded to this environment [45].

The quality of the life of the people is improved by employing AmI through conducting relevant environmental and operational conditions. This can be achieved using adaptive intelligent connection between personalized interconnected systems and services. Ubiquitous, intelligence and context awareness are the major computing areas that are involved in an AmI. This ubiquity forms a pervasive infrastructure where the user is embedded by a large number of interconnected embedded systems. Intelligence systems can include learning algorithms and pattern matchers, speech recognition and language translators, and gesture classification and situation assessment. Finally, context awareness includes tracking and positioning of the all types of objects and finding the relationships between these objects and their environments [45, 154]. AmI are formed to take decisions to benefit the users of that environment based on real-time information gathered and historical data accumulated [15, 136].

Figure 4.1 illustrates an overview of monitoring and interaction system architecture in an intelligent environment representing an AmI. As shown in this figure, the data collected from the sensor network are communicated with a base station and eventually stored in a central database. The communication between the sensor network and the base station could be in either wired or wireless format. More details about sensor networks are presented in the next section.

4.3 Sensor Network

The sensor network has become one of the most important technologies for the 21^{st} century. It consists of spatially distributed independent devices using different sensors to monitor physical or environmental conditions at different locations. Sensor networks have been used in many applications like environmental monitoring building and structures monitoring, military sensing, physical security traffic surveillance, video surveillance, distributed robotics and similar [2]. The main goal of a sensor network is to collect information in an intelligent environment.

Wireless technology for sensor communication is a preferred option, as it is easier to fit wireless sensors in existing homes. However, we do not rule out the use of X10 technology or other well established wired sensor networks in which sensory devices can communicate with the base station via electrical power lines. WSNs, in comparison with wired sensors networks, are more flexible in

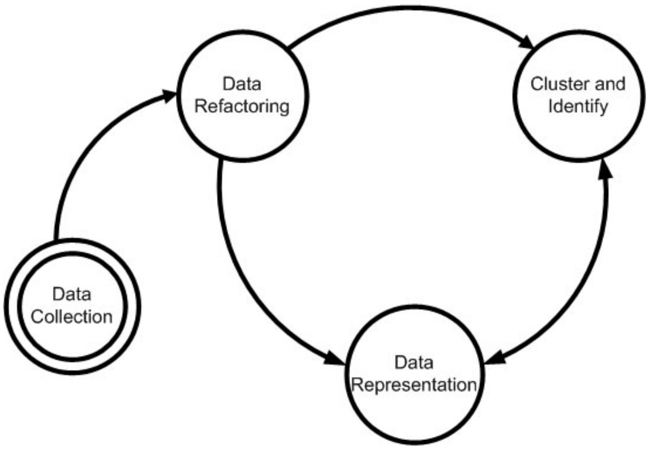


Figure 4.2: Phases in the data handling work flow.

terms of deployment and the required infrastructure of the network in a smart home environment. In WSNs power consumption is the most important concern mainly because all sensory devices are powered by batteries. A system where the occupant was required to change batteries frequently is not ideal. Using either of these two technologies should not make any differences in the results of this thesis. For the sake of simplicity and ease of installation, we have used a WSN comprising movement sensors and door contact sensors.

The sensor data collected from smart environments have different characteristics compared with the ordinary data analysis techniques. Long series of multidimensional data are collected from such sensors which are difficult to analyse and manipulate manually. These data are sometimes noisy if the sensors values are inaccurate or there may be missing values when the sensors fails. In addition, the elements of the sensor data may be a spatial or temporal. The majority of the data source is low-level sensor information which is easy to generate and manipulate. Nevertheless, the challenge, which may be faced when dealing with such low-level data, is the large volume of data collection. For instance, the data collected from motion and light sensors alone are in an average of $10,310$ events per day in the MavHome smart home project [37].

4.4 Data Handling

The data work flow for data handling is illustrated in Figure 4.2. The initial data capture results in a large number of data items which, though time ordered, are not evenly distributed in time, and are initially labelled only by sensor ID, time and Boolean state. The data must then be re-factored so that it can be more easily accessed, enumerated and represented. The data representation and clustering/identification phases feed from each other, as it is only through using different data representations that the separate activities of clustering and abnormal behaviour identification can be easily carried out and assessed.

The sensor data can be represented and mined as sequences or as time series data. These sequences are represented by a series of sensor values. All the sequences are ordered in time and occur sequentially one after another. However, for some applications it is not only important to have a sequence of these events, but also a time when these events occur. Depending on the signal output of the sensor, the time series can be represented either in continuous or discrete values. For example, values gathered from a temperature or humidity sensor will be represented in a continuous series while occupancy is represented in a discrete format.

4.5 Elder's Activities Monitoring

As people grow older, their health gradually deteriorates and more assistance and help is needed in doing their ADLs from their relatives or carer. The activities of daily living such as bathing, toileting and cooking are good indicators of the capabilities of those people. Hence, a system able to monitor the daily activities of the elderly people can play an important role in order to let them to live independently in their own homes. The system should recognize these activities to allow automatic health monitoring and give good guidance for nursing care. This system can also be used to help the people suffering from dementia. For those people, the system could provide a reminding tool about how the activities are prepared step by step. An activity monitoring system consists of two parts: sensors network which is used to collect the data from the environment and a

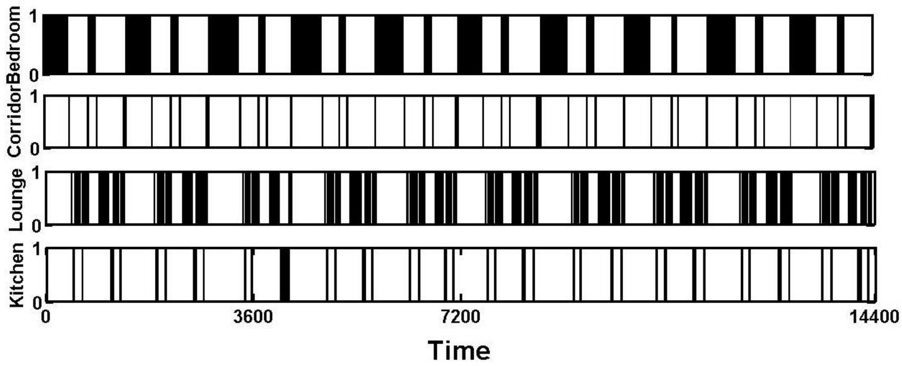


Figure 4.3: A sample of sensor data collected from four motion sensors.

recognition model to recognize and understand the daily behaviour patterns from these sensors [44, 171].

In the research reported in this thesis, WSNs are used to build the activity monitoring system. The daily behaviour patterns of the occupant are then extracted. This information is used to build a behavioural model of the occupant which ultimately is used to predict the future values representing the expected occupancy and other activities.

In this research, the data is captured from occupancy sensors only. This data consists of sequences of an ordered set of movements between rooms in the environment. Sensors record the presence and absence of the occupant in order to track his/her movements in a specific area in the environment. Signals are in a binary format, and they are represented as binary time series. Binary time series extracted from occupancy sensors are usually sparse and contain many repeated constant values. Figure 4.3 illustrates a sample of sensory data collected from four motion sensors.

4.6 Data Collection

In this study, we rely on a data collection system which provides both sensation and transmission. The data acquired includes the occupancy of different areas, environmental attributes, and interactions between an occupant and devices. Sensory devices are responsible for data collection and a variety of sensors are readily available to perform this task. The following list gives the detail of typical sensors [100]:

- Passive Infra-red Sensors (PIR) or motion detectors are sensitive to the movements of living objects. PIR motion sensors respond to changes in heat in the form of infra-red radiation. They are used to identify the movement and then the movement pattern is interpreted as the occupancy. It is important to place PIR sensors in locations where the most effective form of movements is captured. They are normally used to monitor the occupancy of different areas.

- Door/Window entry point sensors are on/off switches which can detect the open and close status of a door/window. Door entry point sensors are relatively reliable as they clearly represent the movement activities.

- Switches contact sensors are same as door entry sensors but they are placed on the fridge and lockers. They can detect the open and close status of these sensors.

- Electricity power usage sensors are used to monitor the activity of electrical devices by measuring their electrical current consumption.

- Bed/sofa pressure sensors are used to measure the presence in and usage of these areas.

- Flood meter sensors are used to provide the states of the taps and flush toilet. Two states are set indicating the cold and hot water taps and flush toilet (when they are opened or closed). They also provide early warning of overflows and leaks.

Only the occupancy sensors including motion and door entry sensors are used to monitor the movement activities of an inhabitant. Data analysis presented in this thesis is based on two real and simulated environments. For real environments, different case studies are presented. Data were collected from either an exiting developed wireless data collection system [6] or using JustChecking system[1] [99].

[1] JustChecking: Supporting independence people with dementia, http://www.justchecking.co.uk

Details of both real and simulated environments are presented in the following sections.

4.6.1 Real Environments

Three case studies are used to validate the results presented in this thesis. In these case studies, each occupant is living alone in different real environments where their movement activities are different from one to another. More detailed description of these case studies are presented below.

4.6.1.1 Case Study I

In this case study, the real data is collected from an environment monitored by JustChecking Ltd monitoring system [99]. The environment is equipped with different sensors with a controller which receives data from the sensors and uploads it to a web-server via an integral mobile phone unit. These are front and back door sensors and lounge, kitchen, bedroom, bathroom and upstairs motion sensors. The collected data from the occupancy sensors were recorded with a resolution of one second to monitor the daily behaviour of the occupant. It should be noted that collected data is based on a single occupant house. Logged data is time stamped and includes sensor ID and a sensor name (type). A sample of raw data collected from this intelligent environment is illustrated in Table 4.1. A software program is used to read the binary data from the actual occupancy sensors and transform them into a time series format. The data is collected for a duration of over one year.

4.6.1.2 Case Study II

The data for this case study is collected from another real environment equipped with JustChecking monitoring system [99], for another elderly occupant. The layout of the smart home environment is shown in Figure 4.4. For this environment, two door entry sensors including front door and back door as well as four motion sensors including kitchen, lounge, upstairs, and bathroom are used. The elderly person was first prescribed some medication, and her health status got worse. Consequently, she was roaming around during the early hours of the day,

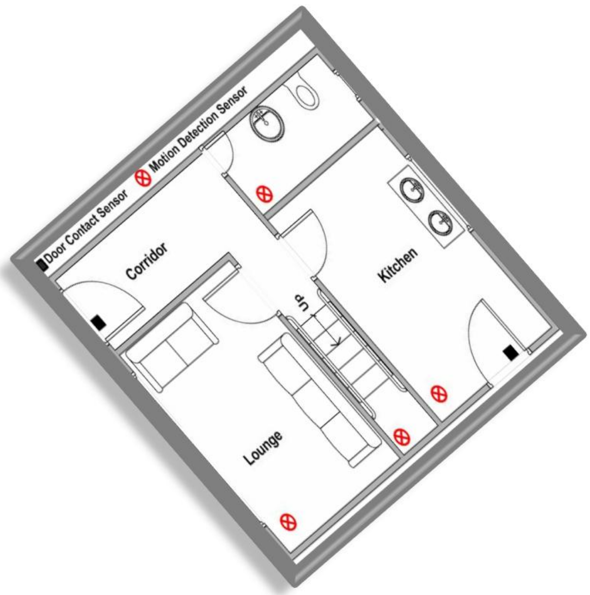


Figure 4.4: Layout of the house and location of sensors of case study II.

and her behaviour was considered as abnormal. Then her first medication was replaced by new medication and the patient's health got better. In this research, the data collected from this environment are split into two separate groups. One group represents the data when the health of the elderly got worse and the other group for the data when the health of the elderly got better.

Table 4.1: A sample of raw data collected from the environment of case study I.

Time Stamp	Sensor ID	Type
21/02/2007 01:15	5	Bedroom
21/02/2007 01:18	5	Bedroom
21/02/2007 01:18	7	Lounge
21/02/2007 01:19	7	Lounge
21/02/2007 01:19	8	Kitchen
21/02/2007 01:19	1	Front Door - open
21/02/2007 01:20	2	Front Door - close
21/02/2007 01:20	8	Kitchen
21/02/2007 01:21	8	Kitchen

4.6.1.3 Case Study III

The data for this real environment is collected from an elderly occupant living in her apartment based on a system developed by [6]. The apartment is located in a council complex in Nottingham in the United Kingdom. The elderly person uses a walker support to help her to move around her apartment where most of her ADLs are carried out. The layout of the apartment of this elderly person with some pictures is shown in Figure 4.5. Four motions sensors covering the lounge, kitchen, bedroom and corridor are used. Additionally, two door entry sensors were used to monitor the bathroom and the main entrance doors. A sample of raw data collected from this environment is illustrated in Table 4.2.

A data acquisition system is installed in the apartment to monitor the occupancy. A wireless receiver agent is used and put in a safe place in her flat. In addition, there are a laptop computer and monitoring portal which constitute a base station for the system. Thus, the collected data by the wireless sensor were transmitted to the base station and logged in a database file using the monitoring portal on the base station. The data is collected for a couple of weeks to monitor the ADLs of the elderly person where holidays and weekends are not included.

Table 4.2: A sample of raw data collected from the environment of case study III.

Time Stamp	Sensor ID:Value	Type
13/05/2008 15:06:13	11:0	Corridor - OFF
13/05/2008 15:06:21	21:160	Main door - Open
13/05/2008 15:08:40	21:0	Main door - Close
13/05/2008 15:08:44	11:160	Corridor - ON
13/05/2008 15:08:46	21:160	Main door - Open
13/05/2008 15:08:46	11:0	Corridor - OFF
13/05/2008 15:08:48	14:255	Lounge - ON
13/05/2008 15:08:49	14:0	Lounge - OFF
13/05/2008 15:08:52	14:255	Lounge - ON

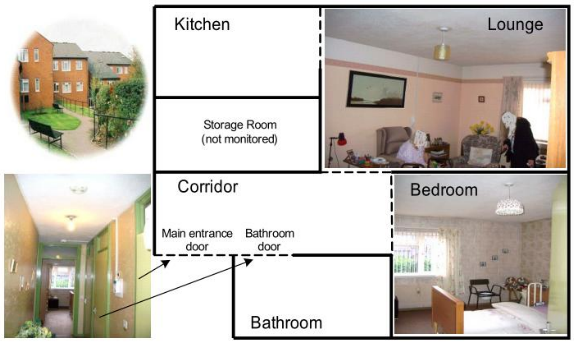


Figure 4.5: Layout of the apartment with some pictures for case study III [6].

4.6.2 Simulated Environment

The environment in the real world is equipped with a large number of intelligent tools such as sensors, actuators and computing components. For many researchers, working with the real environment is rather difficult because the tools in real environment are expensive and collecting data from sensors is one of the main steps in self-adaptive applications. In addition, researchers require large data samples in order to choose and justify better techniques for prediction purposes in intelligent environment. Thus, it would be better to simulate the real environment and generate data without hardware costs [21, 165].

The data generated from this environment was proposed in [5] in which the simulator imitates an occupant's behaviour inside a single living environment by generating an occupancy signal with a resolution of one minute. There are four simulated sensors. The four sensors are: lounge, kitchen, bedroom and corridor sensors. A number of uncertainty factors were used in simulating the movement patterns. It is identified that the simulator developed in [5] does not include some input aspects of an elderly person. Therefore, the original simulator was modified to incorporate trends for different behaviours.

In the next section, the process of adding a trend to the simulator is explained in detail.

4.6.2.1 Modelling Trends within the Simulator

Generally, the analysis of trends is significant for public health observation to identify the health status for a person i.e. if it improves or worsens over a period of time. In addition, it is important for government planning to estimate the future cases in need of health and other services. Trend analysis has been used for a wide range of applications such as forecasting, program evaluation, policy analysis etc. [149].

The simulator developed by Akhlaghinia et al in [5] was modified to include a linear trend on the final signal which emulates the actual data generated from real environments. In the original simulator, the occupancy in a single occupant environment is modelled by using statistical techniques. This model simulates the pattern of occupancy for a single occupant and generates an occupancy signal which is ultimately formulated into a time series. A single occupancy scenario was created to model both the occupant and the environment including the movements and duration uncertainties. However, this system still has to find a way to model the long-term patterns and trends of the occupant activities. Generally, finding the long-term patterns and trends in physical activities is necessary to estimate the progress or deterioration in these activities and also to identify any abnormality within these activities for a medical assistant or caregiver [56].

To model trends within the simulated environment, the simulator needed modification to overcome this limitation. In this case, a linear function was applied on the actual mean duration time to add a long term trend to the final signal. Using the normal distribution of the duration,

$$(\mu - \sigma) \leq \mu \leq (\mu + \sigma) \tag{4.1}$$

is considered to be a normal duration time i.e. no trends (constant) where μ is the expected spent mean time in area and σ is the uncertainty parameters representing the behaviour of the occupant. Then, based on the μ a new mean expected duration time $\bar{\mu}$ is calculated. For example, for increasing and decreasing trend, the following expression is used:

$$\bar{\mu} = \mu + \alpha * t \tag{4.2}$$

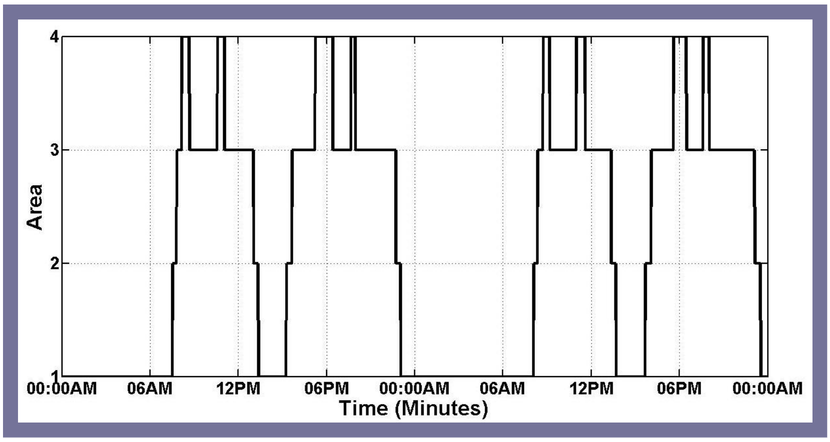


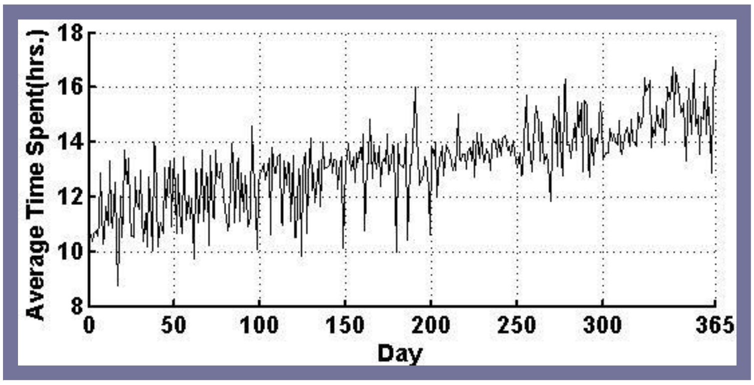
Figure 4.6: A sample of two days of occupancy signal simulation for a four-area environment with no uncertainties in the behaviour of the occupant.

where α is the trend factor or the rate of increasing or decreasing and t is the time. It can be inferred from the above expression that if the slope i.e. α is positive for an area i.e. $\bar{\mu} > \mu$ then the mean time spent by the inhabitant in that area is increasing. On the other hand, the negative slope can result in a decrease in the mean time spent by the inhabitant in that area i.e. $\bar{\mu} < \mu$. The rate of these deteriorations is determined by the amplitude of the slope. In real situation, motion detection sensors can detect these changes. For example, if the monitored occupant is a student and spends more time in a room for studying while approaching the end of university term, this can be detected by a motion detection sensor.

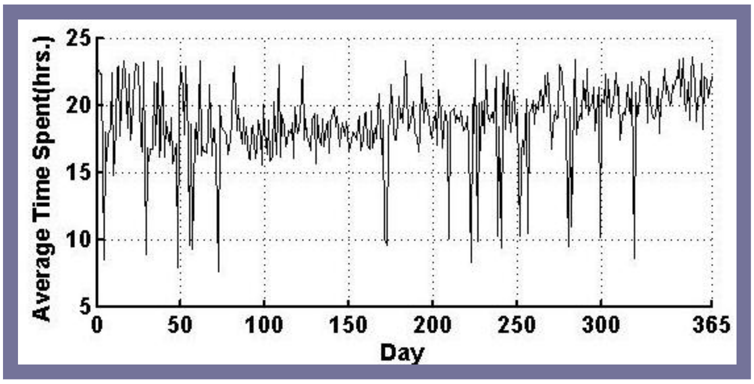
A new profile is added to the original simulator to model trends in the simulated environment, which includes the following parameters:

- Signal type- stable, increasing, decreasing, cyclic and chaotic.
- Trend Factor- slopes amounts for increasing and decreasing trend.

Based on the behavioural modelling presented earlier, an occupancy simulated signal is generated. The simulator generates an occupancy signal for a single occupant in the environment. First of all, a number of parameters should be set to generate a simulated signal. These parameters are: expected occupancy



(a) Without uncertainty



(b) With 5% uncertainty

Figure 4.7: An occupant behaviour of an increasing trend in frequency of data (a) without uncertainty, (b) with 5% uncertainty.

pattern with its mean times, the uncertainty of the occupant's profile, the mean unexpected duration time and signal type together with trend factor.

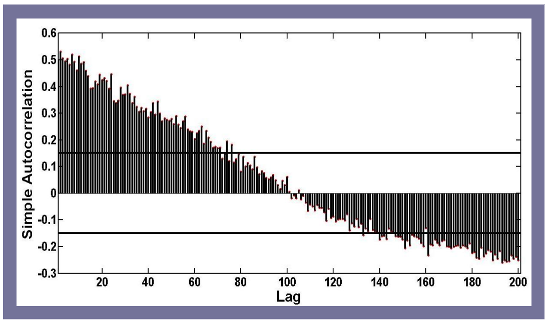
A sample of two days of the occupancy signal simulation for a four-area environment is shown in Figure 4.6. The environment has bedroom, corridor, lounge and kitchen areas. The simulator generates an increasing trend of slope of 0.4 for occupancy behaviour of area 1. It is rather difficult to show the trend using occupancy graph. In this research, a start-time and duration approach which is later explained in Section 5.2.3 to visualize the occupancy sensors in an IIE [51].

To show the effect of uncertainty in the behaviour of the occupant in modelling the trends, a daily average sleeping time for a period of one year is shown in Figure 4.7. The occupancy signal depicted in Figure 4.7-a shows the occupant's behaviour without uncertainty, while the occupancy signal of Figure 4.7-b shows the occupancy signal with 5% uncertainty in his/her behaviour. For instance, an increasing trend in the behaviour is created over time can be seen in Figure 4.7-a. This is a typical of real cases, since there are several reasons that may cause increasing the amount of time the occupant is sleeping such as depression [74]. It can be infered from the other graph with 5%, if the uncertainty is significantly increased, it can be difficult to identify the trend in the signal even though the duration time is changed in non linear manner over time. It is apparent that increasing the uncertainties significantly in the behaviour of the occupant results in randomness in the generated data.

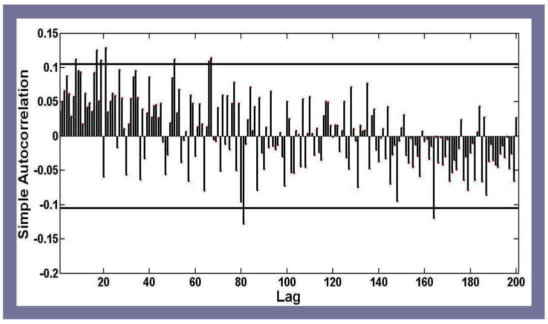
4.6.2.2 Validation of the Simulator

Statistical methods such as temporal autocorrelation plots [130] are used to test and validate the effect of different types of trend in the occupancy signal generated by the modified simulator. The correlation between time-shifted values of a time series is called temporal autocorrelation. In other words, temporal autocorrelation means that there is a strong dependence between the values at a specific time and its past signal values. In this thesis, a temporal autocorrelation plot is used to show the autocorrelation values of the time series generated from the simulated signal.

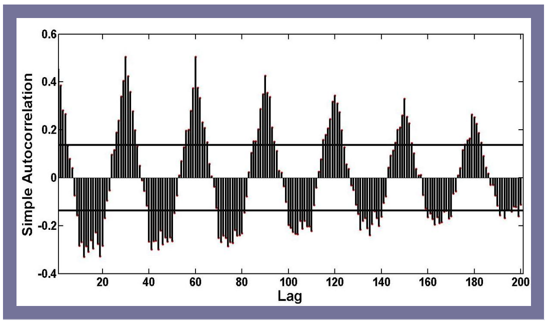
Normally, an autocorrelation plot is used to look at randomness in a data set at varying time lags. The autocorrelation for a random data set is near zero for any or all time lags, while for non random data, one or more of the autocorrelation will be non zero [58, 130]. Figure 4.8 shows the autocorrelation plots of the daily data set generated from the simulator for all types of trends. The simulated data is generated for a period of one year. For increasing and decreasing trends, there is a high dependence between the simulated data. Figure 4.8-a and -b show a strong autocorrelation within the data. In these two graphs, at lag 1 the autocorrelation is high and gradually it is decreased until it becomes negative. Then it starts increasing but in the negative autocorrelation. This is obvious in the occupancy sensors since the occupant may stay at a particular area in the home for an amount of time less or more other areas.



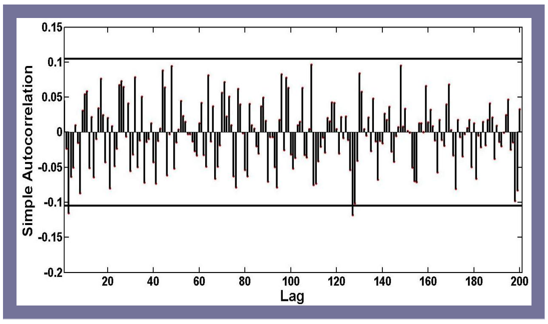
(a) Increasing trends



(b) Decreasing trends



(c) Cyclic trends



(d) Stable trends

Figure 4.8: Auto correlation plots: (a) increasing trends, (b) decreasing trends, (c) cyclic trends, (d) stable trends.

For cyclic trends, the plot shows autocorrelation with a high increasing and decreasing peak in the graph (see Figure 4.8-c) since the correlation between cyclic values is high and as it is move away from the cycle it declines [58]. For no trends or constant signal, the autocorrelation of zero or near zero at any lag means that the data is constant or it is random (see Figure 4.8-d). Otherwise, the simulated signal is considered to be as chaotic.

To show the ability of the simulator in modelling the trend in the behaviour of the occupant, the data generated from the simulator is compared against the actual data collected from real environment monitored (described in Section 4.6.1.2). Figure 4.9 shows the daily behaviour pattern of an occupancy sensor for a period of one year. The increasing trend can be seen in the real data representing the similarity of the generated signal and the occupancy signal in real situation.

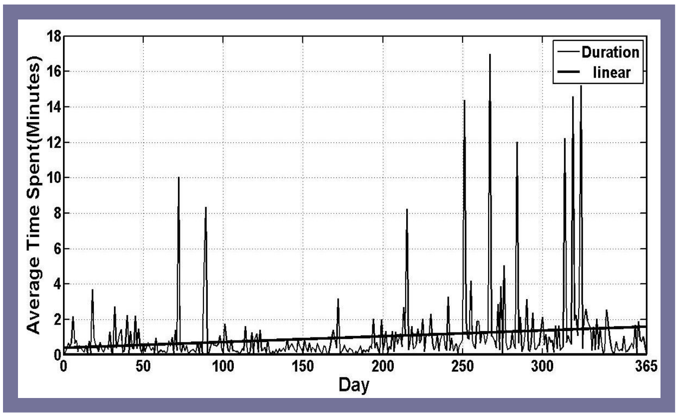


Figure 4.9: An increasing trend in the occupant behaviour generated from a case study for a period of one year.

4.7 Discussions

In this chapter, environment and data collection system employed for our research are discussed. Presented technologies are primarily employed for elderly activities monitoring to help them live independently in their own homes. The difficulties involved in processing the high dimensions of the data collected from low sensor data are also explained.

Real and simulated environments are described in this chapter. Three case studies are created from three different real environments. In addition, for the simulated environment, a simulator is extended to include a linear trend on the output signal which emulates the actual data generated from real environments. The trend of the simulated signal includes an increasing, decreasing, cyclic or chaotic component or is constant (no trend). To discover different patterns (such as increasing or decreasing) and predict future values, predictive models are used using the simulated and real data.

In Chapter 6 and 7, the collected data is used to build a system for automatic health monitoring of elderly people. In particular, they will be used to investigate the data collected from sensor networks.

Chapter 5

Sensor Data Representation and Visualisation

5.1 Introduction

Data collected from an environment representing the ADLs of an individual contain a large volume of complex sensory data. The challenge is to understand human behaviour from these low level sensory data. This could be achieved using common-sense knowledge or using computational intelligence integrated with sensory data. An individual user model can be learned from the sensory data which eventually represents the behavioural model of the user.

Before any data processing, it is essential to represent the data in an appropriate and useful format for behavioural pattern identification and prediction. Also, this process will help to visualize the collected data more effectively. An efficient data interpreting and visualization would help in identification of the daily activities patterns within the sensor data. In this chapter, different data representation and visualization techniques are investigated. Without any loss of generality, the discussion presented in this chapter concentrates on the usage of motion sensors and door entry sensors only.

The layout of this chapter is organized as follows: in Section 5.2, sensor data representation approaches such as start-time and duration, start-time and stop-time are explained. Then, in Section 5.3, sensor data visualization techniques are

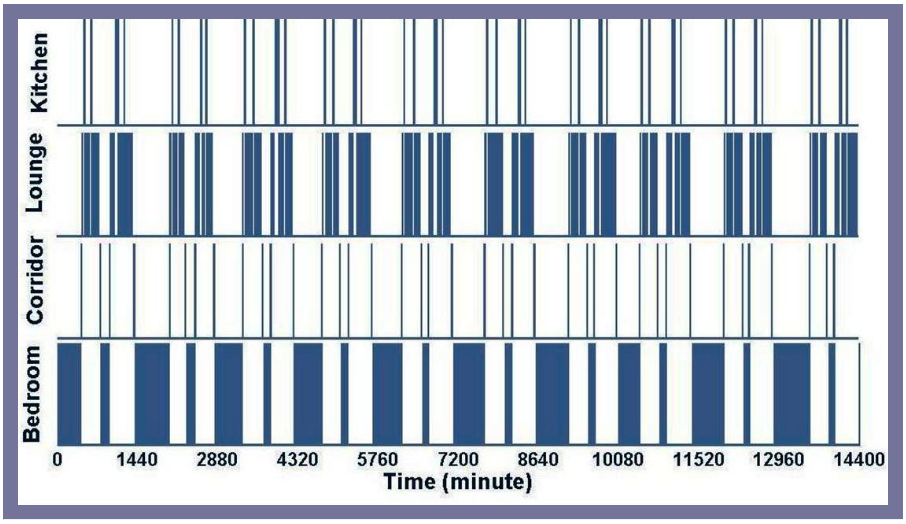


Figure 5.1: Sample occupancy chart for 10 days of data for four different sensors data generated from the simulated environment.

discussed. The conclusions are provided in Section 5.4.

5.2 Sensor Data Representation

The main challenge in an AmI is to find a flexible and useful representation of the large volume of data sets generated from sensors. Many important tasks can be performed with the represented data just as one would do with the original data. These tasks are: clustering, classification, visualization, prediction, identifying patterns and anomalies. The raw sensory data is often difficult to understand. This becomes even more complicated when sensory data from multiple sensors are gathered. Due to the fact that only one occupant is present in the monitored environment, there is no parallel activity in different areas to be detected. Similar patterns can be obtained for a home with multiple occupants. This is achieved by using RFID or other tagging devices to tag different users in the environment. A detailed analysis is reported in [9].

To identify the frequent and abnormal behavioural patterns of a user, we need to collect sufficient data of daily activities to be able to establish the correlation between different events and activities. Furthermore, a trend analysis of the information could be obtained if sufficient data are available. It should be noted that sensor data are collected approximately every second and when this frequency of data collection are repeated for multiple sensors, we would be facing the difficult challenge of interpretation of large amounts of sensor data. To illustrate the complexity of the sensor data, Figure 5.1 shows the occupancy signals from four PIR sensors over a sample of ten day period. It is rather difficult to interpret the data represented in this format.

As stated in Section 2.3, different methods in the literature are used to represent and interpret the ADLs of an occupant living in an IIE. In this chapter, the following methods are proposed and investigated to deal with the binary time series collected from occupancy sensors in an IIE:

- High to Low Frequency

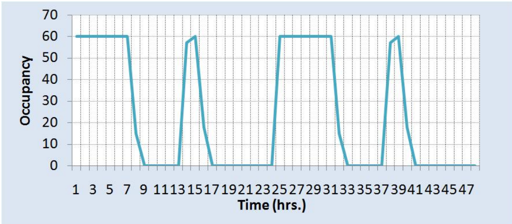
- Start-time and Duration

- Start-time and Stop-time

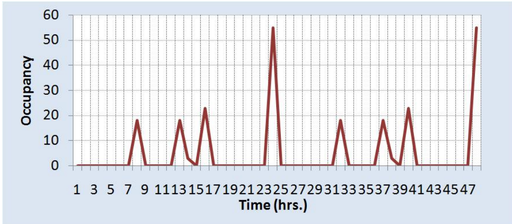
These methods are proven to be useful to summarise the data [51]. Also, they are tested with multiple binary sensors (occupancy sensors, door entry sensors, etc.). Normally, the data extracted from those sensors are usually sparse and contain many repeated constant values. The above methods are expanded in the following sections.

5.2.1 High to Low Frequency

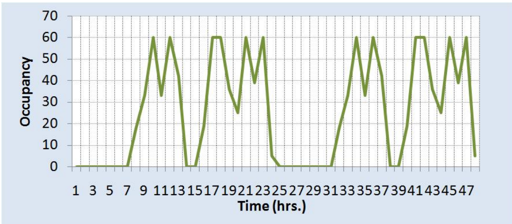
This method converts the higher frequency recorded data into lower-frequency observed time periods e.g. data recorded in seconds converted into hours; daily data into monthly data. Under these conditions, the actual time series data are accumulated over time to form a new, less frequent time series, whereby the collected data is highly reduced, thus helping with visualization. The method for accumulating the transactions within each time period is based on a particular choice of statistics. For example, the sum, mean, median, minimum, maximum, standard deviation, and other statistics can be used to aggregate the transactions within a specific time period. Figure 5.2-a to Figure 5.2-d show the plots of a sample of two days for the four sensors data generated from the simulated



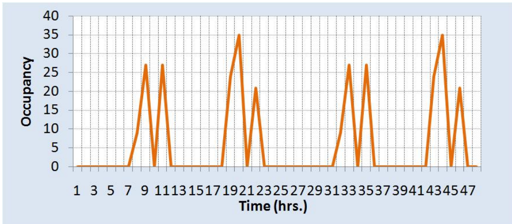
(a) Bedroom



(b) Corridor



(c) Lounge



(d) Kitchen

Figure 5.2: Plots of 2 days of the sample data for the four sensors data generated from the simulated environment using high to low frequency.

environment using high to low frequency. In these figures, summation is used to convert the sensory data from higher frequency (seconds) to lower (minutes) frequency. This technique does not change the fact that the series is still in a sparse format.

5.2.2 Start-time and Stop-time

Start-time and stop-time method is a conversion of the series where the binary time sequence is represented as a compressed time-sampled sequence of starting time and stopping time for each activity. This form of data representation will be used as pre-process data before any modelling or prediction. This form of representation is formally defined in the remaining part of this section. Consider a binary series, $s(t)$, representing the occupancy in a specific area for $t = 1, 2, \ldots, N$, where $s(t) \in [0, 1]$. This signal has two states of 'on' and 'off' representing the presence and absence from a specific area.

$$s(t) = (1, \ldots, 1, 0, \ldots, 0, 1, 1, \ldots) \tag{5.1}$$

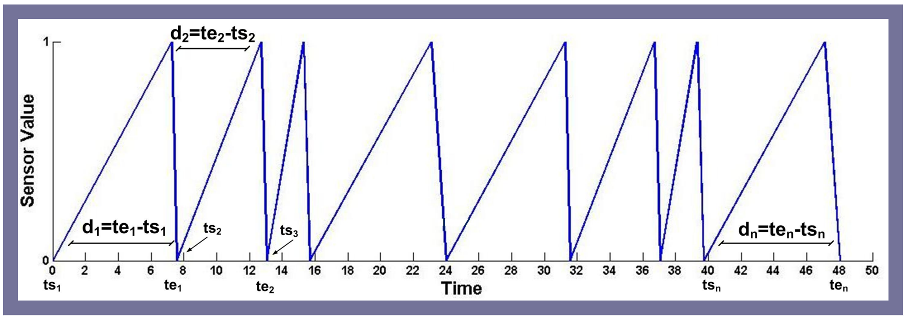


Figure 5.3: Start-time and stop-time conversion of a binary time series extracted from a motion sensor data generated from the simulated environment.

To have a more efficient form of presentation for the binary series, it is recommended to convert $s(t)$ to a start-time and stop-time signal, $x(t)$, as stated below:

$$x(t) = (t_{s_1}, t_{e_1}, t_{s_2}, t_{e_2}, \ldots, t_{s_i}, t_{e_i}, \ldots t_{s_n}, t_{e_n}) \tag{5.2}$$

where t_{s_i} and t_{e_i} are the start-time and stop-time of any event which has resulted in a value of 1 in $s(t)$. The new series $x(t)$ in terms of the start-time and stop-time has fewer values comparing with the long sequences in $s(t)$. Figure 5.3 shows an example of using the start-time and stop-time conversion of a binary time series extracted from an occupancy sensor. This particular conversion shows an accurate and more flexible representation of binary series.

5.2.3 Start-time and Duration

In this method, the binary time series is represented by calculating the start-time and the duration of an event. This form of binary data representation is less frequently used in the field of binary time series. We found that this form of data representation is more suitable for the binary data collected from IIEs [51]. For example, we use the start-time that the person enters a room and duration that he/she stays in a specific location [51].

Start-time and duration method is used when a binary signal is converted into two separate sequences of real numbers representing the start-time and duration of each activity respectively. It should be noted that there is a dependency between these two sequences. Considering the binary series represented in Expression 5.1, start-time series, $y(t)$, and duration series, $d(t)$, are represented respectively as:

$$y(t) = (t_{s_1}, t_{s_2}, \dots, t_{s_i}, \dots t_{s_n}) \tag{5.3}$$

$$d(t) = (t_{e_1} - t_{s_1}, t_{e_2} - t_{s_2}, \dots, t_{e_n} - t_{s_n}) \tag{5.4}$$

This form of representation will make sure that the time dependency and correlation within each component of the series is not lost.

Table 5.1 demonstrate a sample of observations representing the time that the patient spent in the bedroom at each day for a duration of three days. From this sample data, a sleeping pattern is easily identifiable. The graphical representation of these attributes shows significant information about the daily movement behaviour of the occupant.

The start-time and stop-time form of conversion has proved to be effective for modelling and prediction, while the start-time and duration form of conversion has proved to be more suitable for binary signal visualisation. The start-time and stop-time series, $x(t)$ is normalised to a range between 0 and 1 or -1 and 1 before it is applied to any network for modelling and prediction. The normalised signal $x(t)$ will be used to represent the occupancy behaviour in an area.

5.3 Sensor Data Visualisation

Data visualisation has the potential to help in understanding and recognising large volumes of data and also detect patterns and anomalies that are not obvious

Table 5.1: Start-time and duration of a sample of bedroom sensor.

Start-time (hours)	Duration (hours)
13:01	2:20
23:23	10:20
14:06	2:30
23:15	8:15
13:45	2:05

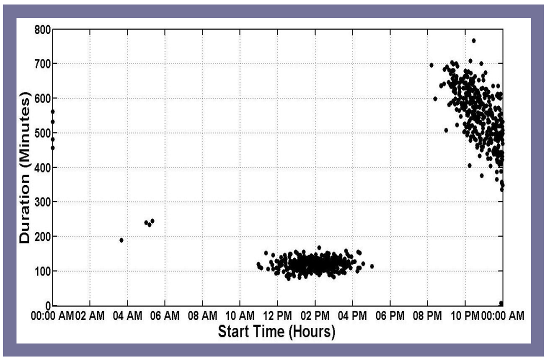
using non-graphical forms of representation. Good data visualisation eases the examination of large volumes of data, and allows deduction to be made from the relationships within the data [124]. In the next section, the techniques which are used to visualize the occupancy sensors data are presented.

5.3.1 Start-time and Duration

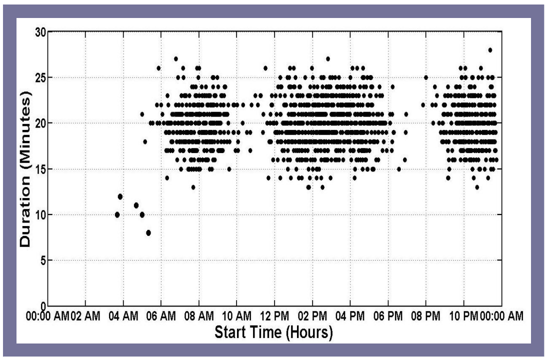
As stated in Section 5.2.3, start-time and duration form of binary sequence conversion helps with the visualisation of the binary series extracted from occupancy sensors. It is rather difficult to track the movements of the occupancy using large binary series. For example, consider Figure 5.1 and Figure 5.4 which show the plots of a sample data set of four sensors generated from the simulated environment for a single occupant. The behaviour of the occupant is more easily interpreted in Figure 5.4 than Figure 5.1. For instance, in Figure 5.4-a, the bedroom sensor plot shows that the occupant always goes to bed at midnight for around 7-10 hours, and he/she usually spends about a two hours period of time in nap sleeping. It is almost impossible to achieve this level of understanding from the raw sensory data represented in Figure 5.1. The bedroom motion sensor shown in Figure 5.4-a represents the projection of the sensory data collected over a long period of time into a 2D graph, collapsing out the axis referring to the individual days. In Figure 5.5 the same activity is illustrated in a 3D graph where start-time and duration of each activity for one year period, with the individual days shown.

Figures 5.6-a and -b show a sample of one month of the real life environment for front and back door sensors data of the environment of case study I. It is rather difficult to follow the daily movement activities from these figures. In contrast, Figure 5.7 clearly shows the activities for a period of over a year. This is most evident in this figure, when the front and back door are opened for rather short time duration and on very rare occasions these doors are left open for long periods. The latter distinctive discrepancy shows that an anomalous behaviour has occurred. We intend to use the start-time and duration form of representation in the rest of this research.

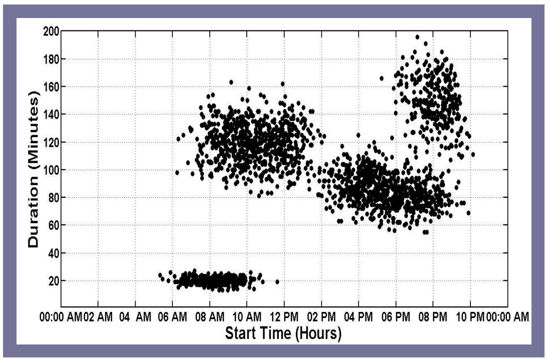
To find the trend within the data and to smooth a time series, a moving



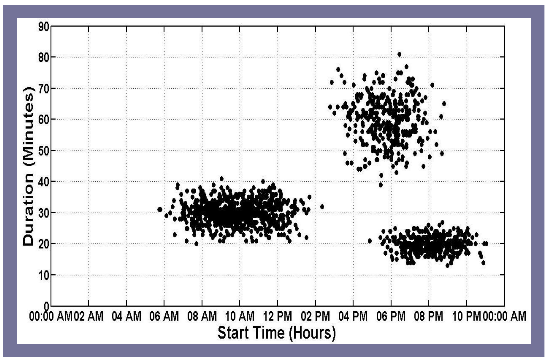
(a) Bedroom



(b) Corridor



(c) Lounge



(d) Kitchen

Figure 5.4: Plots of 365 days of the sample data for the four sensors data generated from the simulated environment.

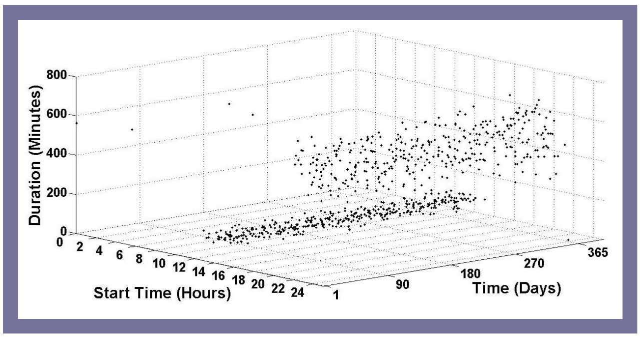
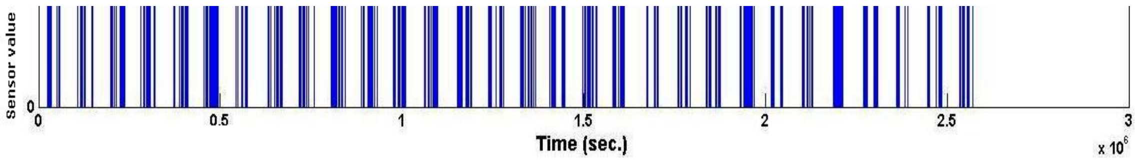
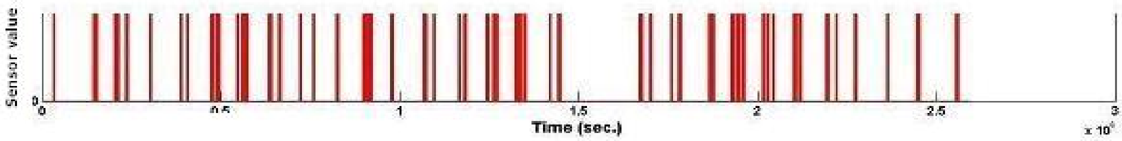


Figure 5.5: Start-time, duration and time of bedroom sensor data generated from the simulated environment for one year sample occupancy chart.

average can be used. Moving average can also be used as a simple prediction method. It is a time series built by computing averages of another set of time



(a) Front door



(b) Back door

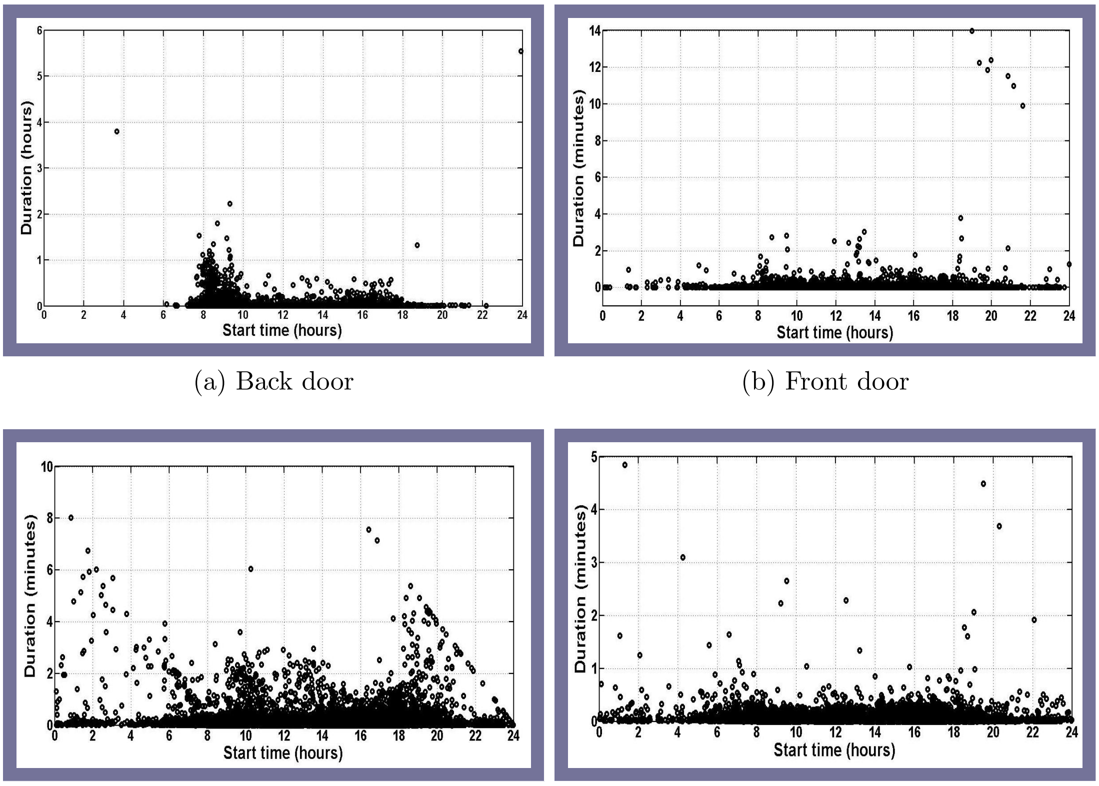
Figure 5.6: Plots of the real data set for door entry sensors in case study I over one month period.

series values [82]. Since the collected data is in a binary time series format, it is quite difficult to highlight the trend. The start-time and duration representation method can identify time series patterns of variable size and gives an effective and efficient representation of the binary time series. For example, Figure 5.8 shows the moving average of the data collected from the same case study over one year (see Figure 5.7) using start-time and duration format.

5.3.2 Principal Component Analysis

Principal component analysis is a statistical tool commonly used to reduce the dimensionality of a data set consisting a large number of interrelated variables, while preserving the variation within the data set [97]. PCA is also a technique which helps in producing a better visualization of the data since it transforms the data in a way that shows the maximum variability within the data [125,135]. More details on PCA are explained in Section 6.5.1.

From PCA analysis principal components (PCs), are identified for the binary data collected in case study I for a period of one year. Figure 5.9-a to Figure 5.9-d show the scatter plots for the first and the second PCs of the binary data collected from the environment for back door, front door, lounge motion and kitchen motion sensors respectively. In these figures, the text numbers are the



(a) Back door (b) Front door

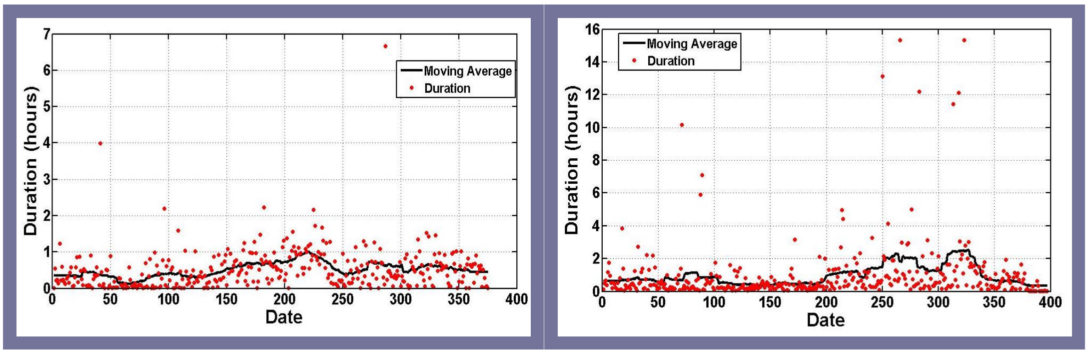
(c) Lounge (d) Kitchen

Figure 5.7: Plots of the real data set for all sensors for a sample of 14 months period.

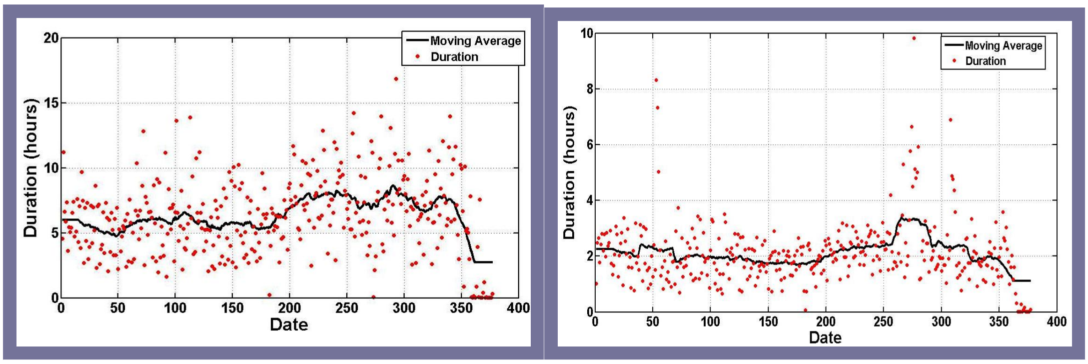
days that activities are carried out.

5.3.3 Data Visualization using Images

Images were also used to visualize the daily behaviour activities of an occupant. Gray scale colour was used, where the white colour represents the sensor value of 1 and black color represents the sensor value 0. The graphical representation of the lounge motion sensor collected from the case study II for an elderly patient is shown in Figure 5.10 and Figure 5.11. The differences in the behavioural patterns between the two groups i.e. before and after taking the new medications are clearly shown. Figure 5.10 show the lounge sensor data for five days before taking the new medicine and Figure 5.11 show the three days after.



(a) Back door (b) Front door



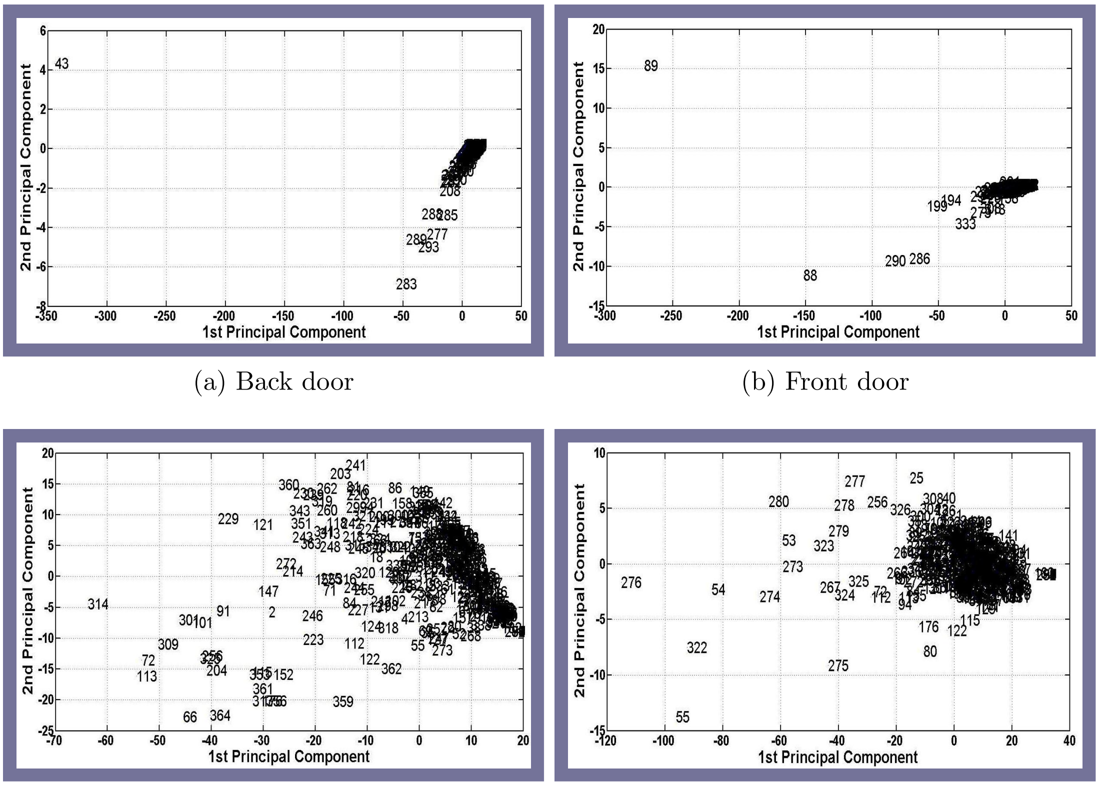
(c) Lounge (d) Kitchen

Figure 5.8: Plots of the moving average of the real data set for all sensors for a sample 14 months period.

5.4 Discussions

In this chapter, some approaches were used to deal with binary sequences such as converting the binary time series values into integer values. An interesting aspect of the proposed approaches is that the data are compressed and still preserve the important features of data. For example in Figure 5.7-a, the original length of data is in order of millions. By visualization these data using start-time and duration, the length of the data becomes only about three thousands. The presented results show that the start-time and duration is the most effective way of representing a large sensor data set. This will also help with the classification of the activities to identify the abnormal behaviour.

In Chapter 7, start-time and stop-time will be used as data pre-processing



(a) Back door

(b) Front door

(c) Lounge

(d) Kitchen

Figure 5.9: Scattered plot for the 1st and 2nd principal components of the data used in case study I.

before any modelling or prediction. Start-time and duration will be used to visualize the results of abnormality identification and pattern prediction.

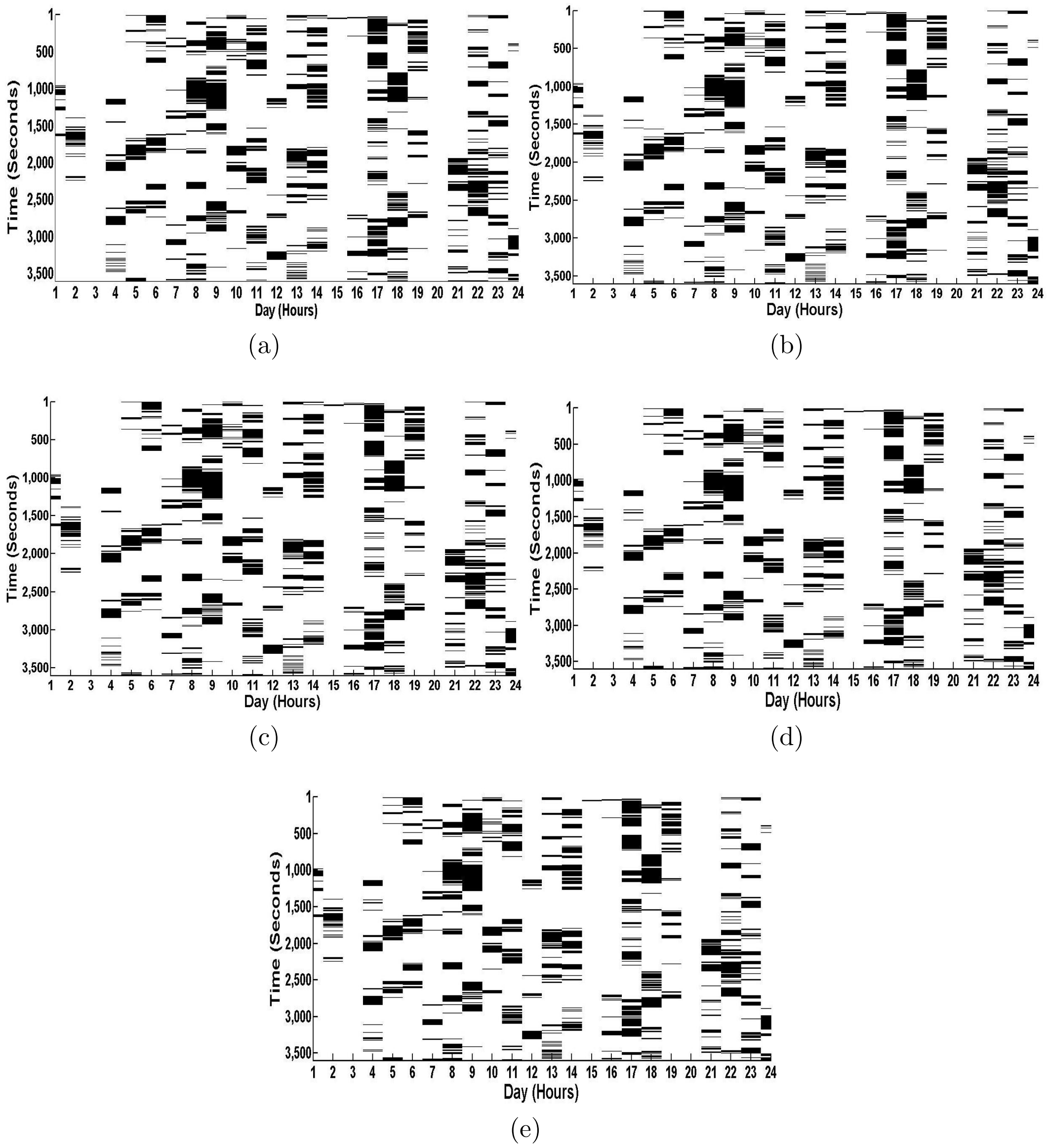


Figure 5.10: Samples of lounge motion sensor data of the environment of case study II for five days which belong to the first group.

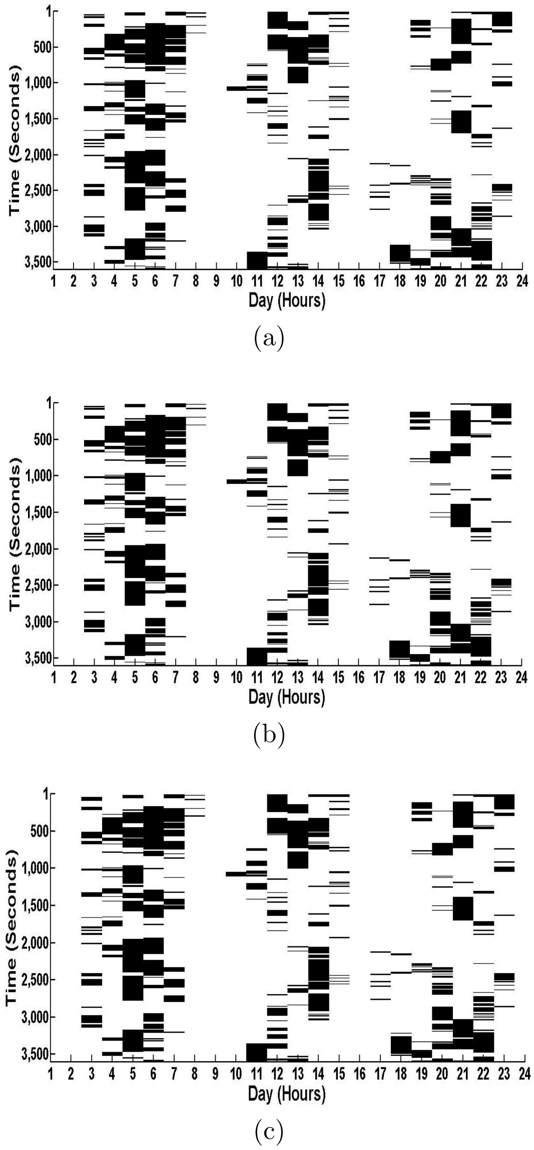


Figure 5.11: Samples of lounge motion sensor data of the environment of case study II for three days which belong to the second group.

Chapter 6

Abnormal Behaviour Pattern Identification

6.1 Introduction

In intelligent environments, it is vital to develop a good understanding of the normal behaviour and distinguish any abnormalities and possible trend in the behavioural changes. The anomaly detection system should apply the limited experience of environmental event history to the rapid changing environment and taking into consideration the temporal relationships between the events. For instance, in a smart home if the refrigerator door is not opened by the occupant during the day as he/she does it normally, then this behaviour is considered as abnormal and an alarm should be sent to the carer. Also, if the occupant turn on the bathroom tap and does not turn it off before going to sleep, then the carer should be informed. If possible, the smart home controller should interfere and turn the tap off [91].

This chapter aims to examine the application of different techniques to identify the abnormality within the behavioural patterns of an occupant in a smart home. In particular, clustering methods, binary similarity and distance measures are used. In addition, an outlier and anomalies detection system is proposed which is an integration of principal component analysis and fuzzy rule-based system. These methods are implemented on data collected from the simulated and real environments as described in Chapter 4 to identify users activities outliers and anomalies.

The rest of the chapter is structured as follows: an overview of anomaly detection behaviour is presented in Section 6.2 followed by an anomaly detection using clustering techniques in Section 6.3. In Section 6.4, the binary similarity and dissimilarity measures are presented to find the degree of resemblance between two binary vectors representing different behaviours. In Section 6.5, an outlier detection system based on PCA and FRBS is explained and the simulation results are also demonstrated. Finally, conclusions are drawn in Section 6.6.

6.2 Anomaly Behaviour Detection

The problem of detecting anomalous/surprising/novel patterns has increasingly attracted attention. Anomaly detection is the identification of previously unknown patterns. The problem is particularly difficult because what constitutes an anomaly can greatly differ depending on the task at hand. In a general sense, an anomalous behaviour is one that turns away from normal behaviour [112].

By monitoring the sensor data, important information regarding any irregular (or anomalies) behaviour will be identified. Anomalies are those odd patterns of data which do not match the normal behaviour. Anomalies can be recognised using different anomaly detection techniques. In many real life applications, these kinds of patterns are also called outliers, discordant observations, exceptions, surprises or peculiarities. Amongst all mentioned terminology, anomalies and outliers are the most frequently used terms within the context of human behaviour detection.

Figure 6.1 shows anomalies within two dimensional data sets representing the sleeping pattern (from bed pressure sensor) of an occupant. Collected data are represented in start-time and stop-time activities format. Most values of the data are in two regions N_1 and N_2 representing night time sleeping and afternoon nap sleeping respectively. These regions are considered as normal. However, the points in region O_1 and points O_2 and O_3 are considered to be anomalies because these points are at different time of the day and different from the normal pattern in the regions N_1 and N_2 [29].

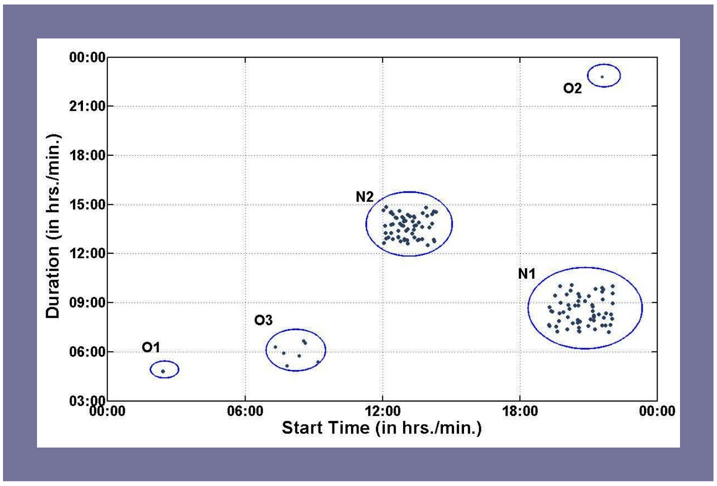


Figure 6.1: Anomalies in a simple 2-dimensional data set.

Outliers and abnormal behaviour identification can help:

- To standardize the datasets collected from sensors,
- To give feedback to the detection techniques,
- To differentiate between the standard data and raw sensor data,
- To raise an alarm and prompt system performance progress,
- To assess the human lifestyles and improve suggestions [91].

In the following sections, user activities anomalies and outliers detection system are investigated.

6.3 Anomaly Detection using Clustering Techniques

In this section, anomaly detection techniques are applied using clustering techniques [111, 112]. Clustering techniques provides a better understanding of data sets. Examples of such techniques are: Self-Organising Maps, K-means clustering, and Fuzzy C-means (FCM) to cluster training data and then use the clusters to classify testing data. Clustering is an important process for condensing and summarising information because it can provide an overview of the stored data. To identify the abnormalities within the sensory data using clustering techniques, the following three categories are identified:

1. The data that reflect the regular or normal data are grouped in clusters, while the data that do not fit in any clusters are treated as anomalies. In this case, any clustering technique can be used and any data that do not find in any cluster are considered to be anomalies.

2. The data that are near their cluster centroid are considered as normal data, while the data that are located far away from their cluster centroid are treated as anomalies. In this case, the data are first clustered and then the anomaly score, which is the distance to its closed cluster, is calculated.

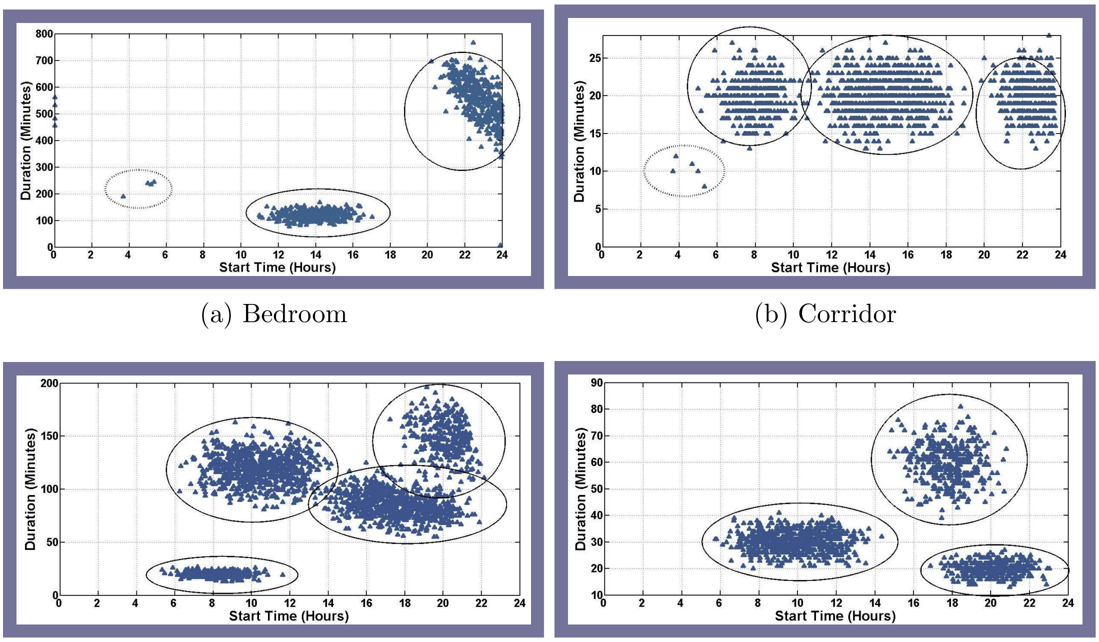
3. The data found in large clusters are considered as normal data, while the small or sparse clusters contain the anomalies. In this case, depending on a threshold value, anomalies can be detected. If the size of any cluster is below this value then the stored data is considered to be anomalous [29,111].

Based on the above categories, the results from using the clustering techniques are demonstrated in the next section.

6.3.1 Results using Clustering Techniques

In this study, start-time and duration data representation is first computed on the binary sensor data. Then, FCM clustering technique is used to divide the data into clusters. The clusters represent the behavioural patterns of the occupant at specific times during a day. Since the patterns represent the movements of the occupant inside the property, there is always one significant activity happening at any time. No matter how many activities happened in that room, at the end only the range of start time and duration that the occupant has spent in a specific room is considered.

An important advantage of clustering technique is that any changes of behavioural patterns (anomalies) from the normal amount of time a person spends in a room can be easily detected. If the daily routine activities are clustered



(a) Bedroom

(b) Corridor

(c) Lounge

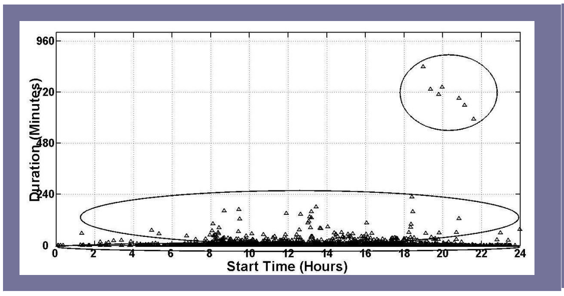
(d) Kitchen

Figure 6.2: Clusters of activities of the simulated data for one year for the four sensors in the simulated environment.

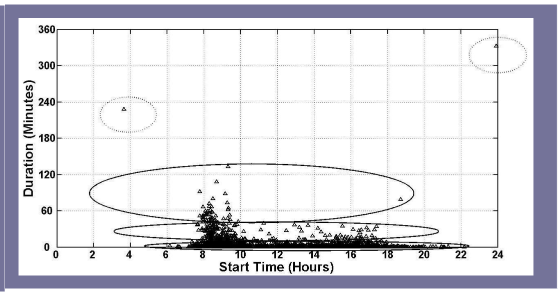
together, then odd activities are then be identified as anomalous behaviour. For example, Figure 6.2 shows the plots of clustering the sensor data for the corresponding Figure 5.4 of the simulated data for a period of one year. In Figure 6.2-a and Figure 6.2-b there are some instances of data that do not belong to any cluster (see the first category above), so these data are considered as abnormal data.

Figure 6.3 shows the clusters for the sensor data set collected from the case study I. A large data sets in a cluster indicates that some similar movement data appear frequently. In contrast, a small data sets in a cluster indicates anomalous behaviour. It is shown that large volume of data can be easily represented, visualised and identified by using clustering techniques.

The number of clusters could vary from one algorithm to another. For example, unlike SOM algorithm, the number of clusters in FCM need to be known in advance. In our experiments the maximum number of clusters was set for supervised algorithms depending on the duration times that the occupant spent in a



(a) Clustered front door sensor



(b) Clustered back door sensor

Figure 6.3: Plots of clustering of the real data set of case study I for only door sensors for a 14 month period.

particular area. In FCM clustering, objects on the boundaries between several clusters do not belong to a specific cluster. They belong to more than one cluster with a certain degree of belonging.

6.4 Anomaly Detection using Binary Similarity and Distance Measures

Binary similarity and dissimilarity (or distance) measures are also used to identify the behavioural patterns for inhabitants in smart homes. The idea behind using such measures is to show to which extent two patterns are similar or not [116]. Similarity measures are defined to measure the degree of resemblance; while dissimilarity measures or distance measures are defined to measure the degree of differences/distance.

Formally, let $A = [A_i]$ and $B = [B_i]$ be two binary feature vectors of the same length, i.e. A_i and $B_i \in [0, 1]$. To compare these two vectors, similarity and distance measures are used. Many binary similarity and distance measures are investigated [27, 53, 62]. Table 6.1 list the five most commonly used binary measures which are used to find the similarity between two binary sequences. The measures excluded the negative matches considering only positive matches and mismatched bits in order to increase their accuracy. In this table, Sim represents the similarity. The term Sim_{11} denotes the positive matches (when both vectors have a value of 1) while the term Sim_{00} denotes the negative matches (both vectors have a value of 0). Sim_{01} and Sim_{10} denote the mismatching bits: Sim_{01} means the first vector has a value of 0 while the second one denotes a value of 1 and Sim_{10} where the first vector has a value of 1 and the second denotes a value of 0 [152].

For each similarity measures defined in Table 6.1, the associated dissimilarity measure is calculated by:

$$Dis(A, B) = 1 - Sim(A, B) \tag{6.1}$$

where Dis denotes the distance. The range returned from these measures vary and to make it comparable, all measures are normalised into the range $[0, 1]$. Similarity and dissimilarity measures have been used in various areas such as information retrieval, image retrieval, chemistry, ecology, psychology, and biological taxonomy, etc. [27, 31, 148].

There are two major distance measures used to find the dissimilarity between two binary vectors. These are:

- The Classical Hamming Distance: The classic Hamming distance can be defined as the number of mismatching bits between two binary vectors of

Table 6.1: Binary vectors similarity measures.

Measure	Sim(A,B)	Range
Jaccard-Needham	$\frac{Sim_{11}}{(Sim_{11}+Sim_{10}+Sim_{01})}$	[0,1]
Dice	$\frac{Sim_{11}}{Sim_{11}+\frac{1}{2}(Sim_{10}+Sim_{01})}$	[0,$\frac{1}{2}$]
Roger Tanmoto	$\frac{Sim_{11}}{Sim_{11}+2.(Sim_{10}+Sim_{01})}$	[0,2]
Kulzinsky	$\frac{Sim_{11}}{(Sim_{10}+Sim_{01}})$	[0,∞]
Anderberg	$\frac{Sim_{11}}{(Sim_{11}+2(Sim_{01}+Sim_{10}))}$	[0,1]

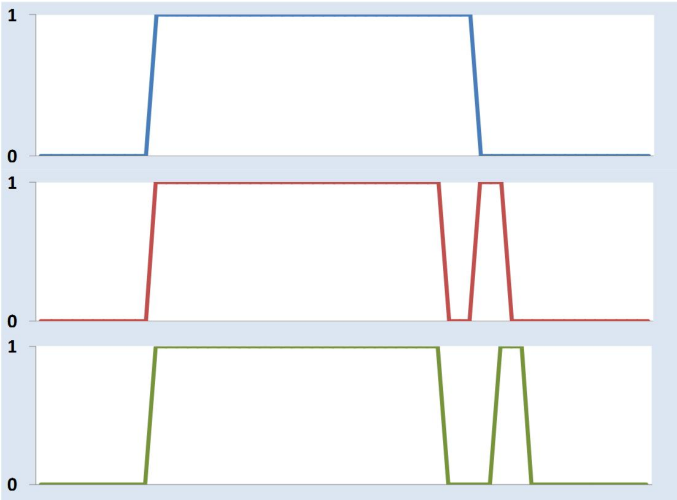


Figure 6.4: Samples of three binary sequences.

the same length. Formally, it can be written as:

$$Dis(A, B) = Sim_{01} + Sim_{10} \tag{6.2}$$

The drawback of classic Hamming distance is that it does not consider the neighbouring bits (the close bits) [20].

- The Fuzzy Hamming Distance: Unlike classic Hamming distance, the fuzzy Hamming distance [108] gives more credit to the neighbouring bits within the binary vector. Besides the number of mismatching bits, fuzzy Hamming distance also measures how far apart the mismatches occur. To compute the edit-distance, a cost function is associated and included three operations [17, 108]. They are:

    - An insertion: $ins(i)$ changes A_i from 0 to 1;
    - A deletion: $del(i)$ changes A_i from 1 to 0;
    - A shift: $sh(i, j)$ changes A_i from 1 to 0 and A_i from 0 to 1

  where A_i is the i^{th} element of A.

To illustrate the difference between these Hamming distance and fuzzy Hamming distance measures, consider the three binary sequences which is shown in Figure

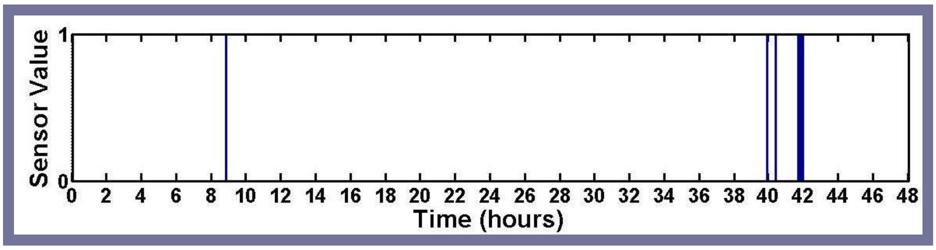


Figure 6.5: A sample of two days of the back door entry point collected from the environment in case study I.

6.4. These sequences can be written as follows:

A : 0000000000011111111111111111111111111111111**111000**00000000000000
B : 0000000000011111111111111111111111111111111**000111**00000000000000
C : 0000000000011111111111111111111111111111111**000**00**111**000000000000

The classical Hamming distance between (A) and (B) has the same distance as between (A) and (C) which is equal to 6. However, (B) is closer to (A) than (C). Fuzzy Hamming distance can measure the distance in terms of closeness to its neighbouring. The fuzzy Hamming distance between (A) and (B) is equal to 4 while the fuzzy Hamming distance between (A) and (C) is equal to 5.5. Hence, (B) is closer than (C) to (A).

Distance measures can be used to find the regular behaviour patterns and eventually any deviation from one day to another. If the result of the distance $Dis(A, B)$ is very small, the two binary vectors are similar. Otherwise, the vectors are dissimilar. For example, consider Figure 6.5 where the back door sensor in case study I is closed most of the time (bit state 0), and it is rarely opened (bit state 1). In this case, it is necessary to choose the index that take into consideration the mismatching bits only and ignores the negative matches (i.e. 0 bits).

Some results are presented in the next section where dissimilarity or distance measures are applied to the data collected from the real and simulated environments. The results are demonstrated to show how these measures can be used to find the unexpected patterns in an occupant′s behaviours.

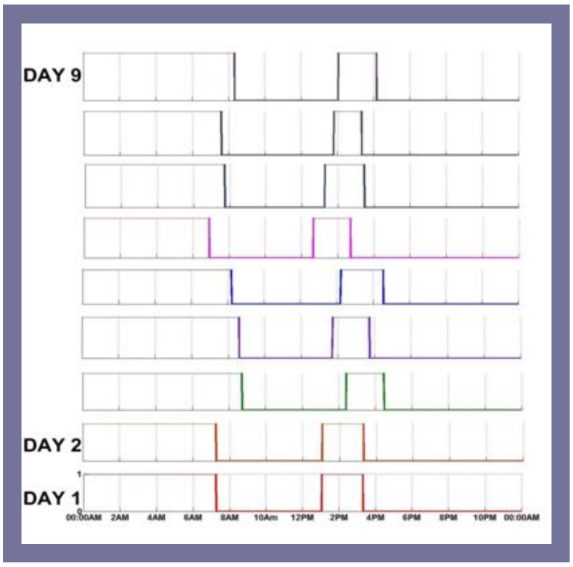


Figure 6.6: Activities of daily living generated from a bedroom motion sensor data for the first 9 days.

6.4.1 Results using Distance Measures

To asses the binary distance or dissimilarity measures, experiments were conducted on the occupancy sensor data collected from the simulated and real environments. In this section, using distance measures the degree of differences between two binary vectors is found and eventually identifying the occupant behavioural pattern. It is important, to know how close or far the sequence of activities in one day are with sequence of activities of another day of an occupant behaviours. For example, Figure 6.6 illustrates the sequences of ADLs for an occupant generated from the simulated data for nine days. Distance measure for nine days and the new data (i.e. the 10^{th} day) is computed. The dissimilarity matrix (%) between the 10^{th} day and all the previous nine days using Jaccard-Needham is shown by the following matrix:

$$D(10 \times 10) = \begin{bmatrix} 0 & 0 & 32 & 21 & 28 & \mathbf{13} & 8 & \mathbf{11} & 25 & \mathbf{14} \\ 0 & 0 & 32 & 21 & 28 & \mathbf{13} & 8 & \mathbf{11} & 25 & \mathbf{14} \\ 32 & 32 & 0 & 13 & 6 & \mathbf{42} & 26 & \mathbf{25} & 9 & \mathbf{22} \\ 21 & 21 & 13 & 0 & 15 & \mathbf{32} & 14 & \mathbf{13} & 9 & \mathbf{11} \\ 28 & 28 & 6 & 15 & 0 & \mathbf{38} & 22 & \mathbf{20} & 6 & \mathbf{17} \\ 13 & 13 & 42 & 32 & 38 & \mathbf{0} & 20 & \mathbf{24} & 36 & \mathbf{26} \\ 8 & 8 & 26 & 14 & 22 & \mathbf{20} & 0 & \mathbf{7} & 18 & \mathbf{10} \\ 11 & 11 & 25 & 13 & 20 & \mathbf{24} & 7 & \mathbf{0} & 16 & \mathbf{5} \\ 25 & 25 & 9 & 9 & 6 & \mathbf{36} & 18 & \mathbf{16} & 0 & \mathbf{13} \\ 14 & \mathbf{14} & \mathbf{22} & \mathbf{11} & \mathbf{17} & \mathbf{26} & \mathbf{10} & \mathbf{5} & \mathbf{13} & \mathbf{0} \end{bmatrix}$$

It can be observed that from above matrix, 10^{th} day has no significant difference from the 8^{th} day while 10^{th} day is quite different from the 6^{th} day. In addition, there is no difference at all between 1^{st} day and 2^{nd} day.

The results for the classical Hamming distance and fuzzy Hamming distance measures are also compared. Experiments on the data collected from door sensors gives almost the same distance for both classical Hamming distance and fuzzy Hamming distance. For instance, the results of the distance measures for the back door entry sensor ine case study I over ten days are shown in Figure 6.7. No significant differences in the results of these measures are shown in this table. Since doors are opened for a short period of time only and it may be reopened again after a long period of time. The results are rather different for the motion sensor data since these sensors are normally having significant changes, which might happen at a close time due to any spontaneous movements. The experimental results on such data shows that fuzzy Hamming distance has performed better than the classical Hamming distance.

Fuzzy Hamming distance gives more credit to the closeness or neighbouring bits in a binary sequence. Thus, this distance results in a lower distance value than the classical Hamming distance. Table 6.2 provides the results of these two measures on the kitchen sensor data collected from the environment of case study I for a period of 20 days. It is apparent from this table that fuzzy Hamming distance has lower distances than Hamming distance since fuzzy Hamming distance gives more credit to the neighbouring bits within these data sets.

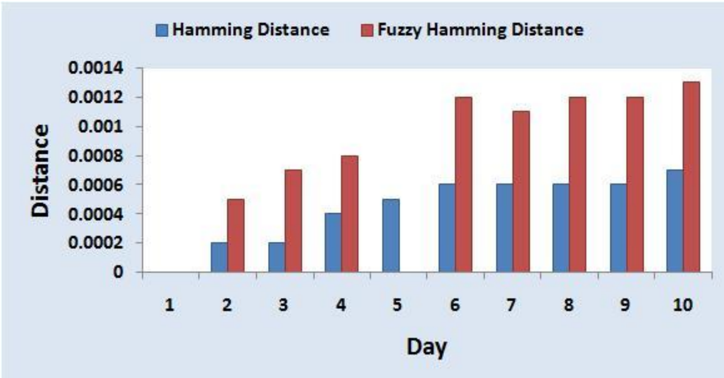


Figure 6.7: Distance Measure for the back door sensor of the real environment of case study I for ten days.

To show the difference two groups of data, another experiment was conducted. A set of data from the motion sensor in case study II are used. To distinguish between these two groups, as with previous experiments, five binary dissimilarity measures were used. Figure 6.8 presents five binary dissimilarity measures on the lounge occupancy sensor in case study II for a period of eight days. The figure shows a significance difference between these two groups. The first five days show the lowest degrees of similarity while the last three days have a highest similarity. As stated in details of the case study II in Section 4.6.1.2, a new medicine was

Table 6.2: Distance Measures for the kitchen sensor data of case study I environment for a period of 20 days.

Day	Classical Hamming distance	Fuzzy Hamming distance
1	1823	1260
2	1735	1240
3	1670	1221
.	.	.
.	.	.
.	.	.
17	387	355
18	286	276
19	146	143
20	0	0

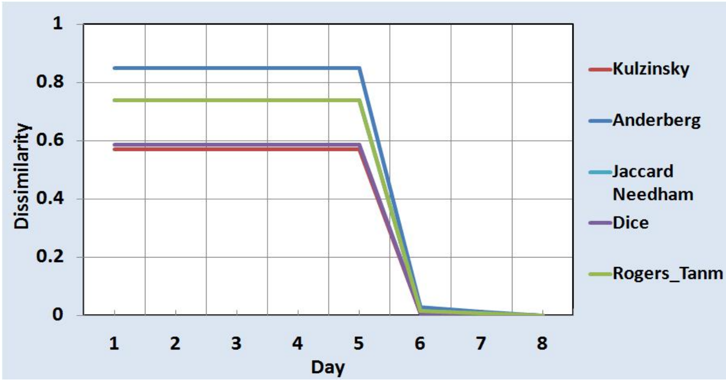
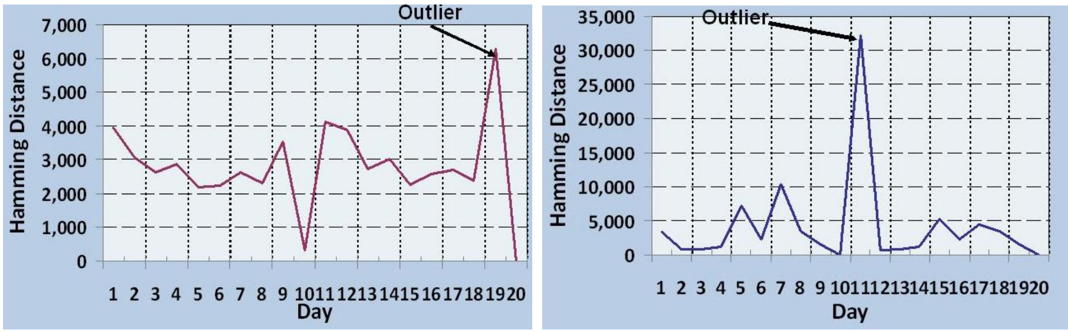


Figure 6.8: Dissimilarity measures for the lounge sensor of case study II for a period of 8 days.



(a) Bedroom (b) Bathdoor

Figure 6.9: Hamming distance measure for the environment of case study III for a period of 20 days.

used by the patient after the first five days. Following three days (day 6, 7 and 8) respectively, the behaviour of the user after new medicine was used.

Comparing the results of these dissimilarity measures with the actual data for both groups, the differences in the behavioural patterns between these groups are clearly obvious (see Figure 5.10 and Figure 5.11 where the lounge sensor data for five days before taking the new medicine and three days after are shown). Overall, the type of sensor data did not affect the results using these binary dissimilarity measures. Also, the experiments based on real environments show that user movements activities, who has some regular patterns within his/her behaviour can be identified using binary dissimilarity measures.

It was also noticed that the anomalies or outliers in one binary sequence of sensor data are not necessarily outliers in another sequence of data for the same case study environment. For example, consider the Figure 6.9-a and Figure 6.9-b represent the Hamming distance measure for the bedroom and bathroom door sensors in case study III for a period of 20 days. As shown in these two figures, the extreme outlier in Figure 6.9-a is in day 19 while in Figure 6.9-b is in day 11.

From the above results, it can be concluded that as the number of days are increased, it is quite difficult to make a decision on how far or how close the new day is with all previous days. Therefore, any significant increase in the dimension of the actual sensor data results in the corresponding increase in the distance matrix which makes it difficult to compare the new data with all previous data. In other words, the distance matrix will become a very high multi-dimensional matrix.

In the next section, a system is proposed to deal with high multi-dimensional data and identify a user activities outliers. The system is also able to distinguish between the normal and abnormal behavioural patterns.

6.5 Anomaly Detection using PCA and FRBS

In this section, the proposed outliers or anomalies detection system is presented. Due to the fact that the data are collected from low level (binary) sensors, which consist of long series of multi-dimensional data, which is usually sparse and contain many repeated values, the proposed system is based on a kind of data reduction to reduce the high dimensionality of the data.

Identifying outliers and abnormality in a high dimensional matrix is a complex process compared with the low dimensional matrix. The outliers in multi-dimensional data do not appear by using each individual dimension i.e. they are not identified using a univariate approach (in this approach outliers are detected when the standardised value of the data point is large). Therefore, it is better to handle the outliers using a multivariate approach where outliers are detected considering all features of the multi-dimensional data. An approach which efficiently reduces the high dimensionality of the data is required for better visualization and ultimately identification of any outlier or abnormality.

There are many methods that can be employed for dimensionality reduction to analysis high dimensional data [39, 71]. These methods can be classified into supervised and unsupervised. These are:

- **Principal Component Analysis :** PCA is an unsupervised approach to understand the main features of data, reducing the dimensionality of multi-dimensional data, finding the relationships between variables of the data, and identifying the trends in the data [125, 135]. PCA has been used in many real-time processes monitoring for detecting changes in operating points, sensor faults, process faults, and plant disturbance [114].

- **Linear Discriminate Analysis (LDA) :** Linear discriminate analysis (LDA) is a supervised approach to discriminate and reduce the dimensionality of data. LDA accomplishes maximum class discrimination through minimizing the within class distance and maximizing the between class distance at the same time [187].

- **Discrete Wavelet Transform:** Discrete Wavelet Transform (DWT) is an unsupervised approach to transform data into another numerically different vector where both vectors are of the same length. By keeping only a small fraction of the important wavelet coefficient, a compressed approximation of data can be kept. DWT is similar to the Discrete Fourier Transform (DFT) , a signal processing approach that converts real-time signals into its components sines and cosines. However, the data compression using DWT is better than DFT since DWT gives an accurate approximation of the original data and needs less space than the [71].

In the first stage of our proposed detection system, PCA is used to find the Principal Components (PCs), and ultimately two error indices that are used to detect changes or outliers within data. Unlike most machine learning and data mining techniques, PCA efficiently reduces the high multi-dimensional data by transforming the data into a new lower dimensional representation.

PCA is able to extract useful features from data set and the variables of the data set and helping in produce a better visualization of the data. It does this by transforming the data in a way that shows the maximum variability within the data. Comparing PCA with other data reduction techniques such as DWT etc., PCA can better process sparse data and is computationally inexpensive [3,71,125,135]. Also, the mathematical foundation of PCA is quite uncomplicated compared with LDA and DWT approaches. PCA is a more appropriate and popular approach than LDA when the data is unlabelled [39].

On the other hand, PCA is a linear transformation approach. Also, the interpretation of its PCs can be difficult which is not acceptable by the domain researchers [39]. To overcome this difficulty, FRBS is used in the proposed system to summarize data and help in classifying outliers and their severity.

In this thesis, an outlier or anomalies detection system is based on a two-stage where the first stage includes a dimensionality reduction of data in which PCA is used. The second stage includes an outlier or abnormality identification using fuzzy rule-based system. The architecture of the proposed outlier detection system is depicted in Figure 6.10. Components of the proposed system to identify the outlier are listed below:

- Collect the sensor data from the environment (described in the previous chapter).
- Calculate the distance matrices for the collected data using distance measure (described in the previous section).
- Conduct the PCA for the distance matrices. The output from the PCA are Hotelling's T^2 and SPE indices.
- Calculate confidence limits for both Hotelling's T^2 and SPE indices. This will be identified as the universe of discourse for the FRBSs.
- Compute the degree of memberships for both Hotelling's T^2 and SPE indices as the inputs of FRBSs.
- Formulate the rules in the FRBSs and check whether inputs exceed the confidence limits.
- Defuzzify the output values to provide a rank for data point to indicate the membership degree of the outlier.

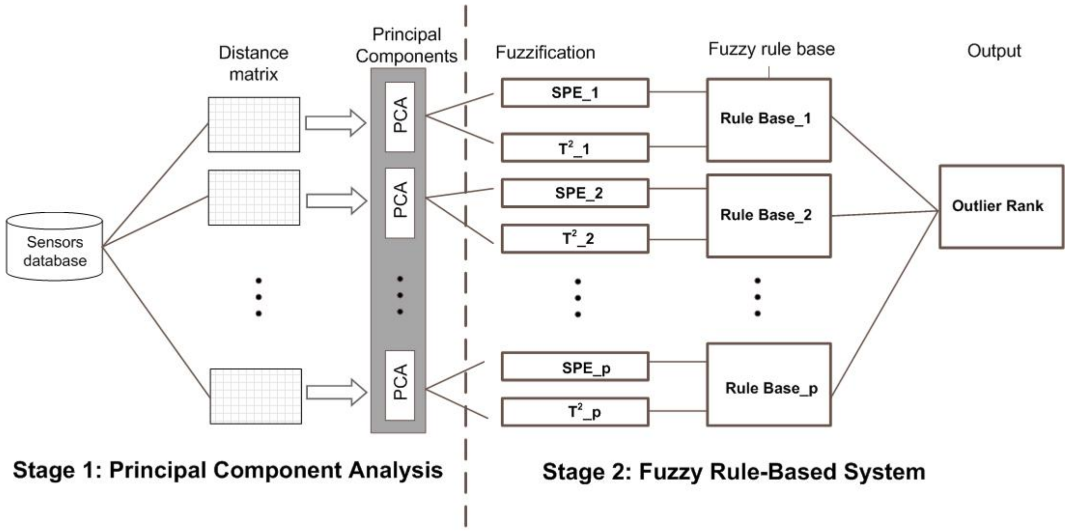


Figure 6.10: Architecture of the proposed outlier detection system.

When new data are available, any outlier or anomalies are detected and ranked accordingly. More details about some components described above are presented in the following sections.

6.5.1 Principal Component Analysis

In this section, PCA is used on the high multi-dimensional distance matrix to find the two statistical error indices Hotelling's T^2 and SPE. These error indices are the inputs to a group of FRBS. Formally, let us consider X as an input matrix which consists of N observations and M variables. Standardization such as z-score is normally used when variables are measured in different units. PCA takes the input matrix and transforms it into two eigenvectors, $e_1, e_2, ..., e_k$, and associated eigenvalues, $\lambda_1, \lambda_2, ..., \lambda_k$, where k is the number of selected Principal Components (PCs). The first PC contains the data with the highest variance while the second PC contain the data with the next highest variance and so on for other components. The format of the eigenvalues and eigenvectors are as follows:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_M \end{bmatrix} \tag{6.3}$$

$$V = [e_1, e_2, \dots, e_k] \tag{6.4}$$

changes within the data are detected using two statistical index measures named Hotelling's T^2 and Square Prediction Error (SPE). When PCs are identified, the T^2 and SPE indices measures are computed. These two statistic measures are briefly described below.

6.5.1.1 Hotelling's T-Squared Statistic

Hotelling's T-Squared (T^2) measures the squared norm of the current sample from the centre of the normal data points region [114]. In other words, the T^2 index measures the variations in the PCs and it is calculated using the following expression:

$$T^2 = X^T V \Lambda^{-1} V^T X \; or \tag{6.5}$$
$$T^2 = \sum_{i=1}^{k} \frac{t_i^2}{\lambda_i^2}$$

where t_i is the i^{th} element in the vector $t = V^T X$. The limit of T^2 index with a confidence level α is:

$$T^2_{lim} = \frac{k(N-1)}{N-k} F(k, N-k, \alpha) \tag{6.6}$$

where the $F(k, N-k, \alpha)$ corresponds to the probability point on the F-distribution with $(k, N-k)$ degrees of freedom and confidence level α.

6.5.1.2 Square Prediction Error Statistic

The Square Prediction Error (SPE) index measures the projection of the data points on the residual subspace [114]. It is calculated using the following expression:

$$\begin{aligned} r &= X^T - X = X^T - V_k V_k^T X \, and \\ & SPE = r^T r \end{aligned} \tag{6.7}$$

The residual matrix r captures the variations in the observation space spanned by the PCs associated with the M smallest singular values. The sub spaces spanned by X and X^T are called the score space and residual space respectively [22]. The limit for SPE index which denotes the upper control limit for SPE index with a confidence level $\alpha = 95\%$ is [114]:

$$SPE_{lim} = \theta_1 \left[\frac{C_a h_0 \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{6.8}$$

where $\theta_i = \sum_{j=k+1}^{m} \lambda_j^i$ and $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$.

The above two indices are checked whether they exceed their control limits. If both T^2 and SPE exceed their upper limit, the process is considered as abnormal and an alarm could be raised. Usually, T^2 is more sensitive to the changes or variation in a process over time that lead to move the process far from normal. T^2 is a measure of the deviation in the model subspace. On the other hand, SPE is a measure of the deviation in the residual space. SPE is applied to identify when the current operation deviates from the normal in terms of parameters that are not dominate [106].

6.5.2 Fuzzy Rule-Based System

In this section, a Fuzzy Rule-Based System (FRBS) is presented. FRBS can efficiently model the vague or uncertain sensory data and can deal with complex data collected from different sensors [120]. It can report the user's activities in terms of linguistic variables instead of the raw or pre-processed sensor data.

In addition, it can give justifications for that report to be more understandable to the end user of the system. FRBS are used in many applications including: automated diagnosis, control systems, image processing and pattern recognition. [66, 123, 124].

In this research, a group of FRBSs is used to classify and summarize users activities of elderly people in an IIE. The following steps are used in developing FRBS:

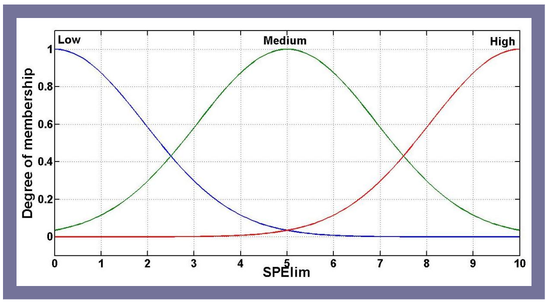
A) Fuzzification

In this step, sensor data set is converted from its crisp value into fuzzy value by assigning the membership degrees of each value in the input and output data set [123]. For instance, the features extracted from the PCs, which is explained in Section 6.5.1.1 and 6.5.1.2, are used to determine the degree of memberships of each sensor data set. These features are represented by the two error indices Hotelling′s T^2 and SPE. These indices form the input variables to FRBSs where the universe of input variables are limited to T^2_{lim} and SPE_{lim} as calculated in Eq. 6.6 and 6.8. The confidence limits for both SPE and Hotelling′s T^2 are computed for each sensor data set. Hotelling′s T^2 and SPE are labelled on a numerical scale based on these confidence limits.

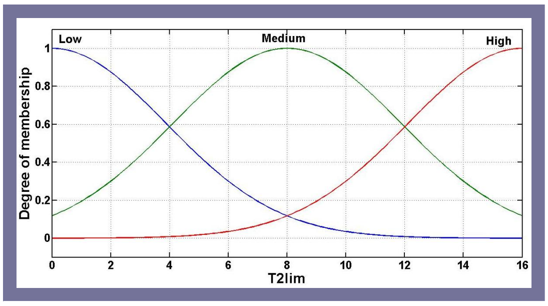
The input variables have three fuzzy membership function μ: $\mu(Low)$, $\mu(Medium)$ and $\mu(High)$. They have different values depending on the kind of the sensor where. The output variable of the FRBS representing by outlier rank is the degree of belief towards the class. Fuzzy sets are defined on this linguistic variable in a similar way as done in case of linguistic variables for the T^2 and SPE inputs. Five membership functions (MFs) are created: μ(Extremely Outlier-EO), μ(Slightly Outlier- SO), μ(Medium-M), μ(More or Less Normal- MN) and μ(Normal- NO). These membership functions are created for each sensor in the environment. The universe of output variable is [0,1].For example, Figure 6.11 shows the membership labels for the inputs and output for the back door entry sensor data set.

B) Fuzzy rules and inference system

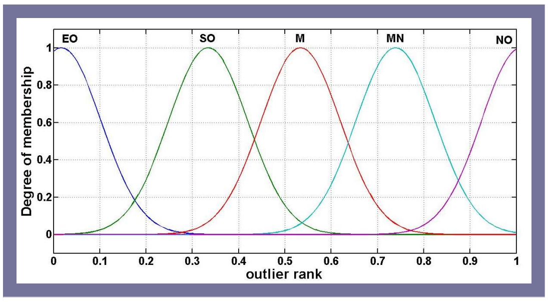
Fuzzy rules are formed in an IF-THEN format where IF part of the rule is called



(a)



(b)



(c)

Figure 6.11: Membership labels for input and output variables for the back door sensor. (a) SPElim, (b) T2lim and (c) Outlier rank

the antecedent and the THEN part is called the consequent. Linguistic variables are used in constructing the rules [123]. The fuzzy inference system with fuzzy output is illustrated in Figure 6.12. The inputs of the fuzzy system have either crisp or fuzzy values, but the output has fuzzy values. In this work, a type of fuzzy rule called Mamdani rule is used as the output of the inference system [94]. The antecedents of the rules are T^2 and SPE and outlier rank is the consequent. Based on the granulated fuzzy labels for each sensor data, fuzzy rules system are

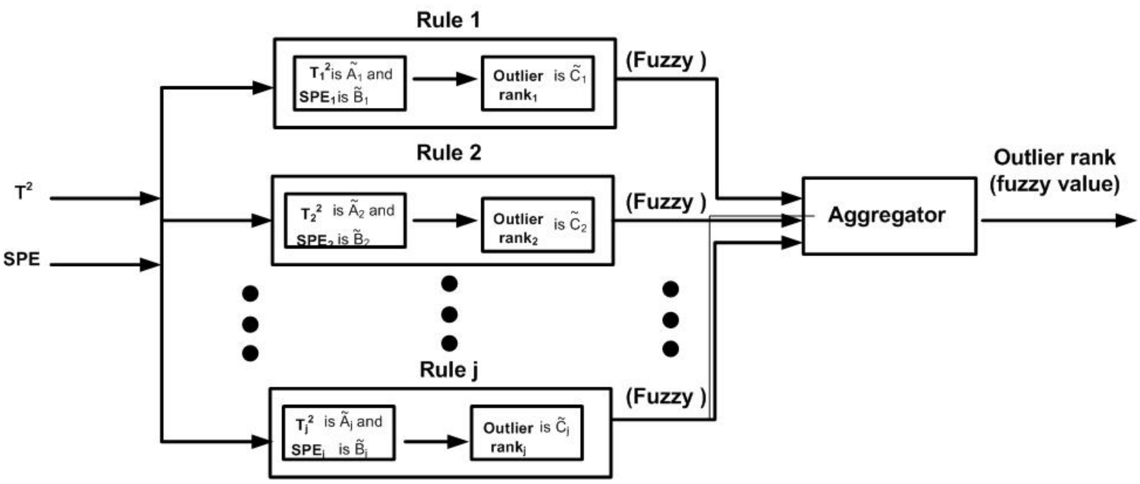


Figure 6.12: Fuzzy Inference system with two inputs and one output.

defined.

For each FRBS, there are nine rules with three membership for each inputs and five membership for each output ($m = 3$ and $n = 5$). The following configuration is employed:

$$R^i_j : If\ T^2_j\ is\ \tilde{A}^i_j\ and\ SPE_j\ is\ \tilde{B}^i_j\ then\ outlier\ rank_j\ is\ \tilde{C}^i_j \tag{6.9}$$

where R^i_j is the label of i^{th} rule for the sensor j. T^2_j and SPE_j are the inputs for the sensor j. $outlierrank_j$ is the output, $\tilde{A}^i_j$ and $\tilde{B}^i_j$ ($i = 1, 2, ..., m$ and $j = 1, 2, \ldots, p$) are fuzzy labels (fuzzy values) for inputs and $\tilde{C}^i_j$ ($i = 1, 2, ..., n$) is the label for outputs. p is the number of sensor data set, m is the number of labels for input membership functions and n is the number of labels for output membership functions.

For each sensor data, fuzzy rules are built using the values of the two statistical error indices T^2 and SPE. If both T^2 and SPE measures exceeds their control limits then the status for the process is abnormal. Otherwise, the status of the process is considered as normal when both indices are less than their limits [114]. It is also possible to reach other options if none of the above are satisfied. Therefore, to reach a decision based on the values of the indices, a fuzzy rule-based system is used to provide the decision. Nine fuzzy rules for outlier rank identification are defined as shown below:

R1: IF T^2 is $\mu(High)$ AND SPE is $\mu(High)$ THEN outlier rank is $\mu(EO)$.
R2: IF T^2 is $\mu(High)$ AND SPE is $\mu(Medium)$ THEN outlier rank is $\mu(SO)$.
R3: IF T^2 is $\mu(High)$ AND SPE is $\mu(Low)$ THEN outlier rank is $\mu(SO)$.
R4: IF T^2 is $\mu(Medium)$ AND SPE is $\mu(Low)$ THEN outlier rank is $\mu(MN)$.
R5: IF T^2 is $\mu(Medium)$ AND SPE is $\mu(Medium)$ THEN outlier rank is $\mu(Medium)$.
R6: IF T^2 is $\mu(Medium)$ AND SPE is $\mu(High)$ THEN outlier rank is $\mu(SO)$.
R7: IF T^2 is $\mu(Low)$ AND SPE is $\mu(High)$ THEN outlier rank is $\mu(Medium)$.
R8: IF T^2 is $\mu(Low)$ AND SPE is $\mu(Medium)$ THEN outlier rank is $\mu(MN)$.
R9: IF T^2 is $\mu(Low)$ AND SPE is $\mu(Low)$ THEN outlier rank is $\mu(NO)$.

The final outlier rank is decided based on the rank of the outlier for each sensor. The final rank is calculated as:

$$outlier\ rank = \min_{j=1}^{p} (outlier\ rank_j) \tag{6.10}$$

In the next section, these fuzzy rules are used to classify outliers and distinguish the normal and abnormal behaviour patterns in an IIE. In addition, the severity of outliers are identified.

6.5.3 Results using PCA and FRBS

In this section, the proposed outlier detection system is validated using case studies presented earlier in Chapter 4. In the beginning of our analysis, the binary distance measure is computed on the raw data to construct the distance matrix. Binary distance measures are used to improve the outliers and abnormality identification. A sample of the results obtained for the Hamming distance for the kitchen occupancy sensor for case study I is shown below:

$$D(365 \times 365) = \begin{bmatrix} 0 & 12375 & ... & 10209 & 10094 \\ 12375 & 0 & ... & 13026 & 13291 \\ 9987 & 15734 & ... & 14686 & 14629 \\ . & . & ... & . & . \\ . & . & ... & . & . \\ 6664 & 11561 & ... & 10191 & 11620 \\ 10209 & 13026 & ... & 0 & 12373 \\ 10094 & 13291 & ... & 12373 & 0 \end{bmatrix}$$

where $D(i,j)$ represents the distance between day $i : 1, ..., 365$ and day $j : 1, ..., 365$. i and j are the indices for different days and it should be noted that $D(i,i) = 0$. It is almost impossible to visualise and identify normal and abnormal behavioural patterns of the occupant from this matrix.

To select the important features for the above distance matrix, PCA is applied. The question is how many PCs should be retained for analysis? There are many ways to choose; one way is to select only those whose Eigen value is greater than one. Another way is to apply Cumulative Percent Variance (CPV) approach. It can be calculated using the following formula:

$$CPV(a) = \frac{\sum_{i=1}^{a} \lambda_i}{trace(R)} 100 \tag{6.11}$$

CPV is a measure of the percentage of the variability ($CPV(a) \succ 90\%$) captured by the first principal components. A graphical plot, called scree plot, is also used to choose the number of PCs. In this graph, all the Eigen values are plotted in their decreasing order and the knee in the graph is identified. To select the components, count the PCs between the knee and the high component [97]. The results of the scree plot are shown in Figure 6.13 based on the kitchen motion sensor data. The first two PCs were selected. In this thesis, scree plot are used to choose the number of PCs to keep in the model. Only the PCs that have Eigen values greater than 1 were selected since component with Eigen values less than 1 have less variance than did the original value.

PCs are used to calculate two indices namely residual SPE and Hotelling′s

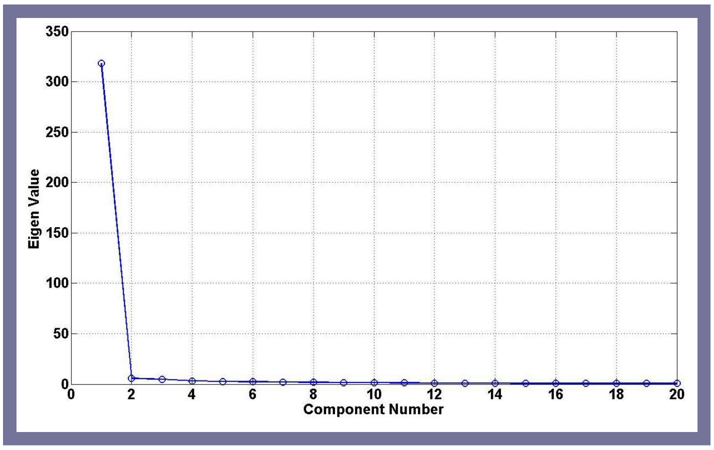
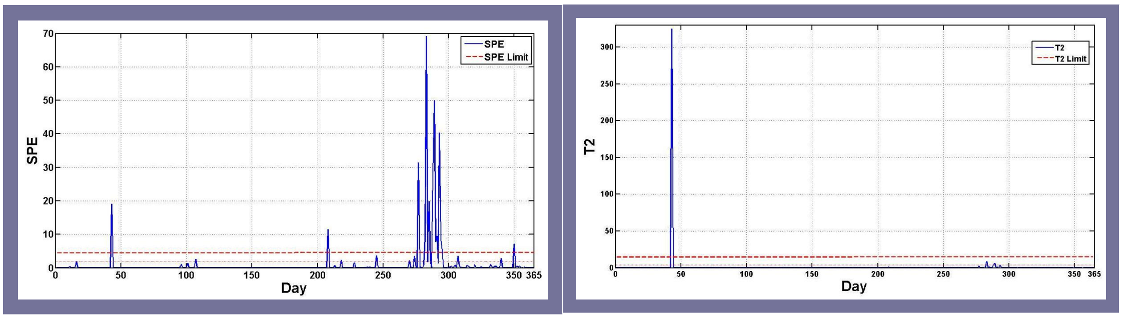


Figure 6.13: Scree plot for determining significant number components for PCA analysis based on data extracted from the kitchen motion sensor for one year.

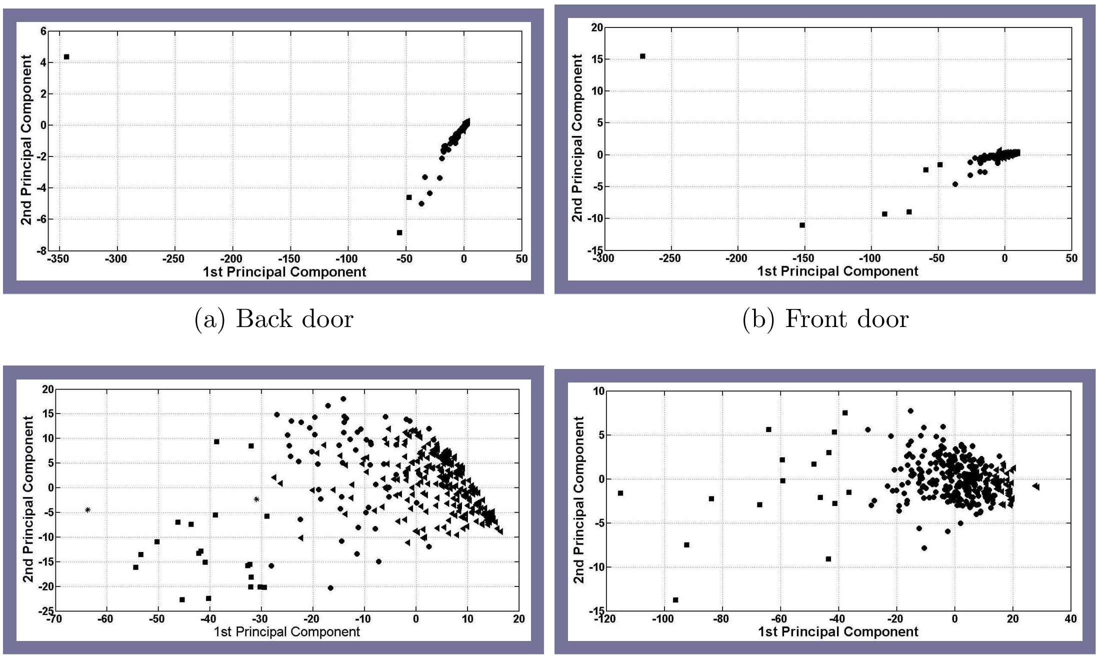


(a) (b)

Figure 6.14: Principal component analysis statistic measures for the back door entry sensor data (a) SPE index, (b) Hotteling's T^2 index.

T^2 measures. The SPE and Hotelling′s T^2 measures for the back door sensor in case study I is shown in Figure 6.14. These measures are used in the second stage of the process to classify outliers within the data sets. These indices are the inputs to FRBS classifiers with one output representing the outlier rank. For case study I, the SPE limits are 1.9255, 6.2451, 211.0567 and 20.9023 for back door, front door, lounge and kitchen sensor respectively. The T^2 limit is 3.8671 for all sensors since the most variations are located on the 1st and 2nd PC and T^2 index limit depends on these two PCs. Based on the granulated fuzzy labels for each sensor data, fuzzy rules are defined.

By using the fuzzy rules, the outliers and anomalies are identified along with

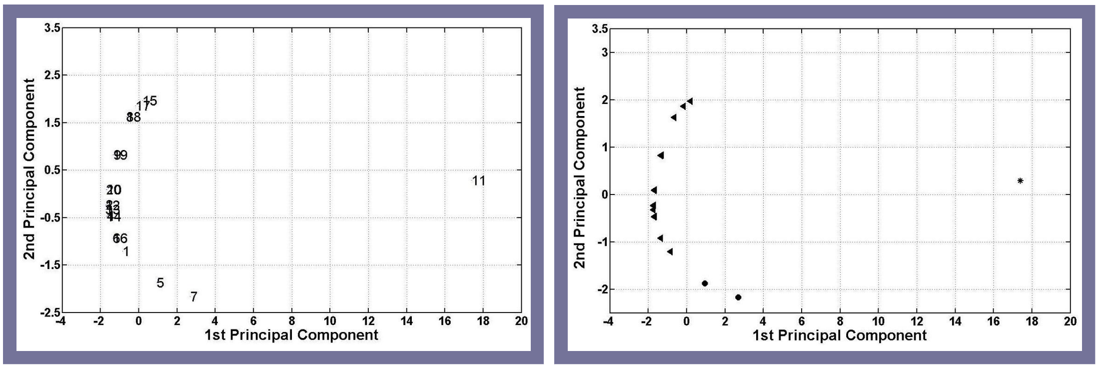


(a) Back door (b) Front door

(c) Lounge (d) Kitchen

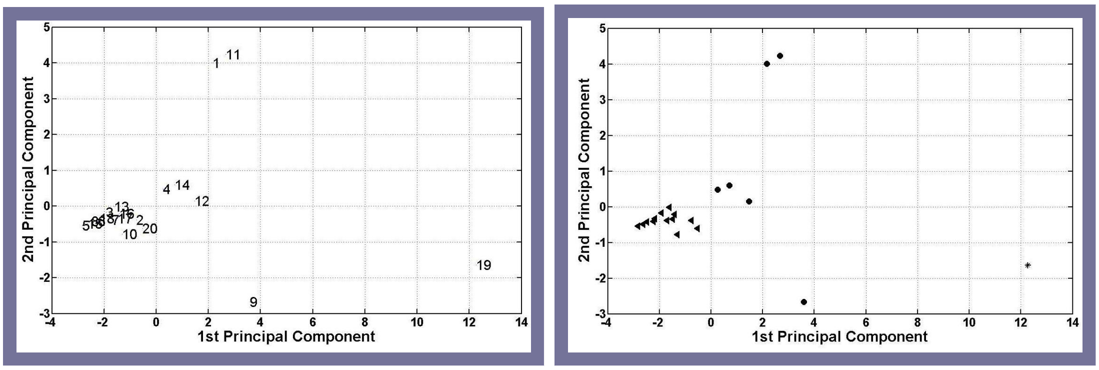
Figure 6.15: Scattered plot for the 1st and 2nd principal components of the data used in case study I with classification. Triangles represent normal, squares represent extreme outliers, stars represent slight outliers and circles represent more or less normal pattern.

the relative position of the outliers within the data (outlier rank). In Figure 6.15-a to Figure 6.15-d labelled PCs based on the fuzzy outlier rank are shown for the same results shown in Figure 5.9-a to Figure 5.9-d. It is obvious that the outliers are detected according to their position in the actual binary data sets. In addition, the most frequent observations (normal group of data) were selected from similar ADLs carried out during the year. However, a few observations, which are located far from the normal group of data, represents set of an individual days where ADLs are unlikely to be carried out (abnormal group). For example, three clear groups of data are recognized in Figure 6.15-b, these are: (1) Normal group (in fuzzy label NO and representing by triangles) represents set of the days in the year where the back door is not often opened and closed, (2) Abnormal or outlier group (in fuzzy label EO and representing by squares), which is located to the left of the graph (i.e. day index 43), represents the data where the back door



(a) Without classification (b) With classification

Figure 6.16: Scattered plot for the 1st and 2nd principal components of the bedroom motion sensor data used in case study III. Triangles represent normal, squares represent extreme outliers, stars represent slight outliers and circles represent more or less normal pattern.

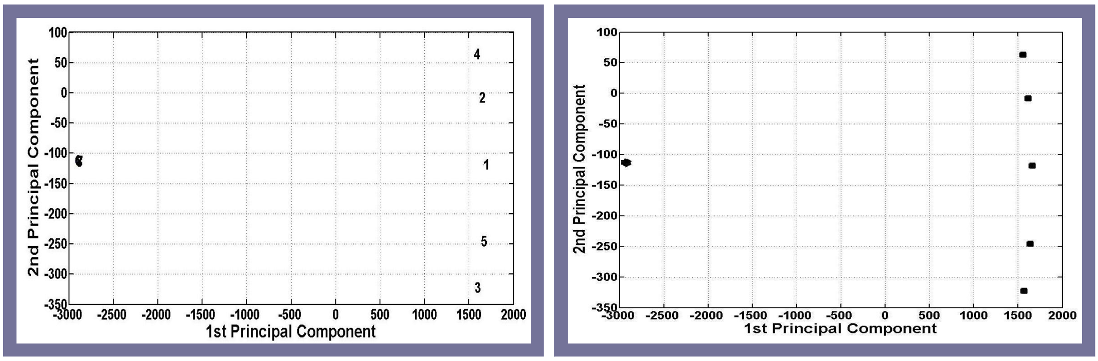


(a) Without classification (b) With classification

Figure 6.17: Scattered plot for the 1st and 2nd principal components of the bathroom door data used in case study III.

remained open for a long time, and (3) More or less normal group (in fuzzy label MN and representing by circles) represents the set of the days where the door is frequently opened but for a short time. It can be concluded that the proposed outliers system gives a visual grouping to the data of similar features and detects outliers and anomalies.

If there are slight outliers and changes or extreme outliers, alarm messages could be sent to the carer to help the elderly person. Figure 6.16-a and Figure



(a) Without classification (b) With classification

Figure 6.18: Scattered plot for the 1st and 2nd principal components of the front door data used in case study II.

6.17-a show the scatter plots for the first and the second PCs of the binary data collected from the environment of the bedroom motion sensor and bathroom door entry sensor of case study III. The corresponding labelled PCs based on the fuzzy outliers rank are shown in Figure 6.16-b and Figure 6.17-b. As shown in these figures, no extreme outliers are found in these data sets, there are only slight outliers (stars) and more or less change in her behaviour (circles). Therefore, in these cases an alarm should be sent to the carer to check if there is something wrong.

Our experiments were repeated for data collected in case study II. Figure 6.18-a shows the scatter plots for the first and the second PCs of the binary data collected from the environment of the front door entry sensor of case study II. The corresponding labelled PCs based on the fuzzy outliers rank are shown in Figure 6.18-b. There is a significant difference between these two groups. The first group shows the normal data (circles) since their SPE and Hotelling′s T^2 are less than the confidence limits, while the second group shows the extreme outliers or anomalies (squares) as their SPE and Hotelling′s T^2 are greater than the confidence limits. The results obtained from this real environment has shown that proposed outlier detection system can effectively distinguish between the normal and abnormal data.

6.6 Discussions

In this chapter, different similarity and distance measures are used to investigate the effectiveness of these measures in identifying outliers in data collected from an IIE. It highlights both progressive (similarity) as well as sudden (dissimilarity) changes in behavioural patterns, both of which are of interest in ADLs in a smart home. Finding dissimilarities between two vectors means that both vectors have changes in patterns of attributes. The values that have common patterns seem to be closer to one another than those with different patterns. It has been found that the type of the sensory data is also important to find the distance measures between two binary sequence. For dissimilarity or distance measure, fuzzy Hamming distance gives lower distances than the classical Hamming distance for motion sensors since these kinds of sensors are normally having significant changes.

The integration of principal component analysis and fuzzy rule-based system to the Hamming distance processing system are investigated to identify outliers and anomalies in an IIE. The intention was to determine the effect of both the residual SPE and Hotelling′s T^2 in improving the results of PCA and how they are utilized in the fuzzification process of a fuzzy rule-based system. The proposed system successfully distinguishes between the normal and abnormal or outliers data points. Simulated and real environments including three case studies are used to show the effectiveness of our detection system.

Chapter 7

Abnormal Behaviour Pattern Prediction

7.1 Introduction

Data interpretation helps us to better understand the ADLs and separates the anomalous behaviours when an activity has happened. This would also be useful in generating a report to summarise the activities of the patient over a long period of time. However, this would not help the carer to make necessary arrangements in advance. To improve the proposed system, a predictive method is required to predict the future values of the activities based on the historical data available from activities recorded by each sensor.

This chapter assesses the suitability of the predictive techniques as a solution to the binary time series prediction representing the ADLs of an elderly in an IIE. Different predictive models are used to predict the time series data representing the movements of the inhabitants in their own homes. Prediction is necessary to automate their regular and frequent behaviour daily activities. Several experiments are devised, executed, and evaluated using different ANN techniques. The techniques, which are chosen for prediction, are mainly divided into two groups of time delay and recurrent neural networks and the results are then compared. The investigated techniques included; focused time delay network, layered recurrent network, NARX network, Elman network, echo state network, long short term memory and recursive self-organizing map.

The rest of the chapter is structured as follows: in the next section, the measures that are used to find performance of the predictive techniques are presented. The cross validation methods to evaluate the performance and comparison of the predictive techniques are introduced in Section 7.3. In Section 7.4, the prediction results using HMM is demonstrated followed by the prediction results using ANN techniques and predictive techniques verification. In Section 7.7, a comparison between ANN techniques is discussed. Finally, a summary and discussion is presented in Section 7.8.

7.2 Prediction Performance Measurement

To identify the ability of the prediction model, it is important to choose an appropriate error measure. In this study, depending on the input to the model, whether it is binary or non binary, different error measurements are used.

For binary series, after training the binary time series data, a similarity measurement between the two binary series (i.e. actual and predicted data) is used. As mentioned in the previous chapter, different similarity coefficients are proposed by researchers in different fields. A similarity coefficient indicates the degree of similarity between object pairs. Jaccard coefficient [190] is one of the most commonly used similarity measurement coefficients for binary series. We have compared and investigated this measure in our study. Jaccard's coefficient is a measurement of asymmetric information on binary (and non-binary) variables.

The above error index will be used only for binary series prediction. However, when signals are converted into either start-time and stop-time or start-time and duration, then a continuous error measure is required. We have used Root Mean Square Error (RMSE) as defined below to measure the difference.

$$RMSE = \sqrt{\frac{1}{M}\sum_{1}^{M}(y(n+1) - \hat{y}(n+1))^2}$$

where $y(n+1)$ is the actual value of the time series, $\hat{y}(n+1)$ is the predicted value, and M is the number of data points that network model has to predict.

RMSE is used to compute the differences between the actual observation values and the predicted values which result from a predictive model.

7.3 Cross Validation

In any learning algorithm, the data is split into two sets: training and testing or unseen data. The data is often split into 2/3 for training data and 1/3 for testing data set. The learning algorithms have often an over fitting problem where the algorithm is well trained on the training data and is poorly performed on the testing data set or unseen data. Cross validation is applied to estimate and evaluate the performance of the learning model, to select and tune the learning model performance. Using this method, the data is divided into two sets: training data used to train and learn a model and validation set used to validate the model [147].

There are various methods used for cross validation, these are:

- Holdout cross validation: the test data is holdout and unseen during the training phase i.e. there is no overlapping between the training and testing data sets. One of the disadvantages of using such approach is that all the validation data is not used during training phase and system performance is highly depended on the choice for the training/testing split. Also, the data in testing set may be significant for training and if it holdouts, the performance prediction becomes poor.

- K-Fold cross validation: the data is divided into k equally sized folds. The training and validation are performed in k iterations. In each iteration, k-folds of data is holdout for validation and the remaining folds are used for learning.

- Leave one-out cross validation: the data except one observation are used for training and one instance of data is used for testing. It is a special case of k-fold cross validation.

- Repeated k-fold cross validation: the k-fold cross validation is executed many times so that the method performance estimation and comparison

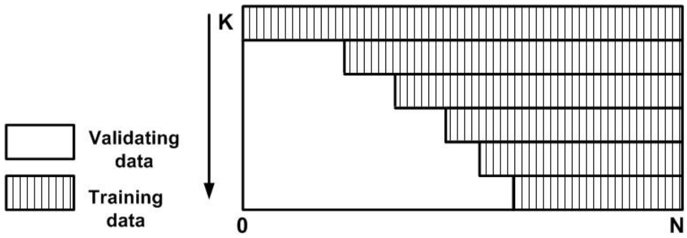


Figure 7.1: Validation procedure schematic. The data is partitioned into k contiguous blocks of training data.

are increased [105, 147].

The experiments in this chapter rely on a special case of validation approach to improve the ANN algorithms generalization and increase performance. Since the temporal relationships (order of data) between the input data is important, the data is divided into k contiguous blocks. In each iteration different starting point for each contiguous block is used for the training data and the remaining contiguous points for validation data. Figure 7.1 shows the schematic of the validation process. In this figure, all values beyond N (where N is the training cut-off) is used for testing and all data point before N is used for training. A contiguous time series data is created for training data set. The starting point for training data is different for each block and ranges between 1 and N-1.

7.4 Prediction Results using Statistical Techniques

Statistical techniques are used in many time series prediction. In this research, HMM predictor (as described in Appendix A) is investigated to predict the time series data collected from sensors network. It is specifically used to predict the movements behaviour activities of an elderly living alone her home. The prediction is based on the start-time and stop-time conversion as inputs to the HMM. Two sets of simulated data are used: the first set of the simulated data represents the binary time series for a very regular occupant i.e. the generated simulated signal has no duration uncertainty. The second set of data represents the binary time series for an occupant with a less regular behaviour i.e. the generated signal has 6% duration uncertainty.

The prediction results in terms of RMSE using basic HMM predictor for the first set of the simulated data for the bedroom, corridor, lounge and kitchen sensors data are approximately 20%, 19%, 18% and 23% respectively. The results for the second set of the four simulated data are approximately 27%, 21%, 14% and 21% respectively. The current study found that HMM has poor performance in predicting the binary and converted input data sets. A possible explanation for these results may be due to the lack of the hierarchy in this kind of modelling, the increase in the length of time series may need large volume of time series runs from HMM, and the difficulties in processing the temporal data from different time scales.

Therefore, it seems that there is a need for a time series prediction model, such as ANN algorithms, which are able to solve the problem of the temporal relationships within a large and complex data is required.

7.5 Prediction Results using ANN Techniques

In this section, several ANN techniques, that are described in Chapter 3, are investigated. These techniques are mainly divided into two groups TDNNs and RNN techniques. They are used for predicting the binary time series data collected from the ADLs of an inhabitant in an IIE. In particular, using these techniques, the next locations of the inhabitants are predicted. The prediction will help to provide us with information related to their health trend and take an advance action.

As mentioned in Chapter 4, the data sets are generated from two different environments. The first data sets are generated by the simulated environment while the second data set contains the real data collected from different case studies. In both environments, the actual data represents a binary time series. The actual data is recorded with a resolution of one minute for the simulated data, and with a resolution of one second for the data collected from the real environments.

Training and testing data set are used in evaluating the prediction techniques as described in Section 7.3. The training data set are used to build the predictive model and the testing data set are used to analysis the prediction accuracy of the model. Experiments are conducted on these data sets. The experiments reported here are based on seven days of training data set and three days of testing data set for the simulated environment. For the case studies data collected from the real environment, 14 days of training data set and 6 days of testing data set. The first $1,209,600$ points (in binary format) are used as training set and the last $518,400$ points for the testing data set.

Most of the selected techniques use only one input unit which is driven by the actual sensor at time t. The output unit is the value of the same sensor at time $t+\ \tau$ (τ is the number of steps ahead for prediction). The number of hidden units is different in each technique. It is also possible to connect all the available sensors at the same time as inputs to these networks and compute the prediction of these sensors readings using the combined data set from the single input prediction. The advantages of using just one neural network for all sensors, instead of using a separate neural network for each of the inputs, are to reduce the amount of memory and computation used. In this situation the number of input and output units depends on the number of sensors.

In the next section, ANN techniques are used to show their performance in predicting the movement activity of an elderly in an IIE. Two sets of experiments were done to show the performance of these techniques. The first set of experimental results focus on the prediction analysis of TDNN techniques while the second set is on RNN techniques.

7.5.1 Prediction Results using TDNNs

For TDNN techniques, experiments are conducted using focused time delay neural networks, layered recurrent network and NARX network. In this section, only the results of NARX network are demonstrated. The choice is made based on the high accuracy of prediction obtained using this network compared with other TDNNs. In these experiments, serial-parallel architecture of NARX network is used for learning the dynamic behaviour of an occupant living in an IIE. In serial-parallel architecture, the actual output, instead of the estimated output, is returned to the input of the network.

To train NARX network, a number of parameters are used. These parameters are: number of input tapped delay D_x, number of feedback output delay D_y, number of neurons in the hidden layer (N), number of epochs time required for training, and learning rate.

The type of inputs for most ANN techniques have either binary or non binary input data set. In this thesis, two sets of input data are analysed; these are binary (discrete) and real (continuous) time series data. For binary input data sets, the prediction results are demonstrated in terms of graphs where the actual sensor values are denoted by (-) and the predicted values are denoted by $(\ldots)$. Prediction results of the converted data are also demonstrated in terms of graphs where the actual sensor values are denoted by (O) and the predicted values are denoted by (+). The prediction results for the converted data (start-time and stop-time) are visualized using start-time and duration. Start-time and duration method of conversion seems to be a practical and easy tool to visualise large binary data collected from sensors network.

In the following sections, the prediction results of NARX networks are evaluated using binary and converted input data sets.

7.5.1.1 Results for Binary Input Data

For binary input data, the input to NARX network is driven by the actual binary time series extracted from the sensor data at time t. The output is the next value of the data, i.e. at the time $t + \tau$. Considering the size of the real dataset we were not able to generate a model based on the actual binary data. Therefore, we only present the prediction results for the simulated data set.

Figure 7.2 shows the NARX network prediction results for a sample of three days of the binary time series data with a very regular patterns. The prediction is based on two hours step ahead i.e. $\tau = 120$. In Figure 7.2, the predicted data is very close to the actual data in terms of durations since these data represent a very regular occupancy behaviour. For instance, the bedroom occupancy signal shows the sleeping patterns for an occupant who goes to his/her bed at a regular time (around $22:00$) to sleep about $6-7$ hours and also takes a nap of about 2

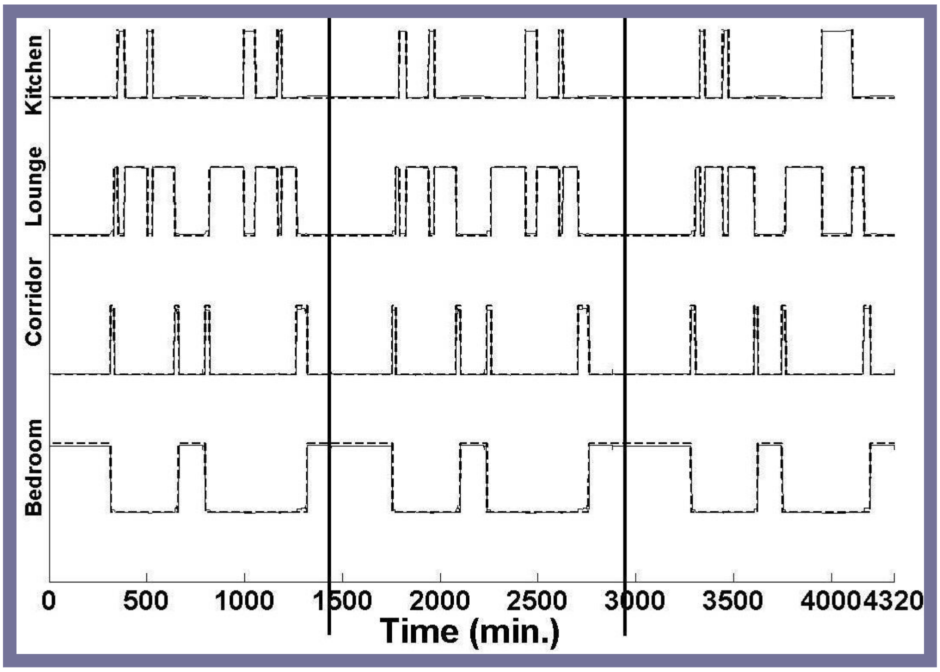


Figure 7.2: Sensor values and two hours ahead predicted values using NARX network for three days sample of simulated activities data for a regular occupant (no duration uncertainty), (-Predicted values; – – – Actual values).

hours after mid day.

The prediction results of the simulated data with some irregular patterns using NARX network is shown in Figure 7.3. From this figure it can be observed that NARX network can successfully predict the irregular occupancy patterns where the occupant spent more time in some area than others in the environment. For example, the bedroom occupancy signal shows that the occupant spent most of his/her time in the bedroom which might indicate an early detection of illness or depression. However, the difference between the predicted and the actual data is barely noticeable. This is slightly less accurate for the corridor sensor. Corridor signal is relatively more chaotic. Our observation is that more chaotic signals are expected to be more difficult to predict.

As stated in Section 7.2, the errors between the predicted and the actual data for the binary data is calculated using some binary similarity measurements. Jaccard coefficient is applied to calculate the similarities between the predicted and actual data sets. For the simulated data with regular patterns, The similari-

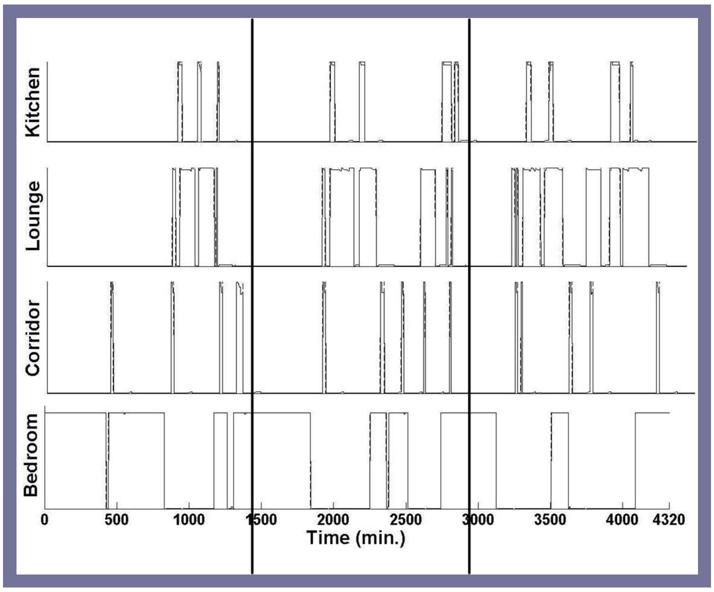


Figure 7.3: Sensor values and two hours ahead predicted values using NARX network for three days sample of simulated activities data for less regular occupant (6% duration uncertainty), (-Predicted values; − − − Actual values).

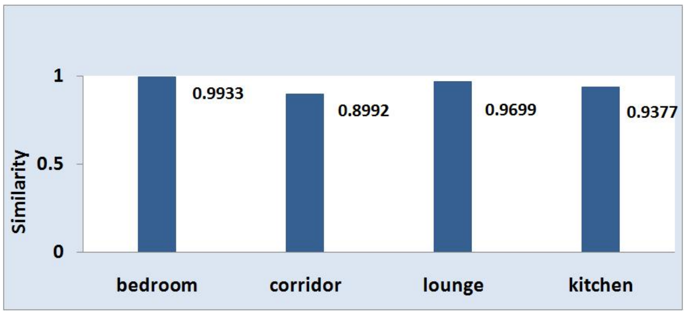


Figure 7.4: Similarity measurement accuracy between the actual and the predicted binary data generated from the simulated occupancy signal of Figure 7.3 using NARX network.

ties have a high values indicating that the differences between the actual and the predicted output of the NARX network are very small. Figure 7.4 shows the similarities between these two binary data sets generated from the simulated occupancy signal of Figure 7.3. The similarity between the predicted and the actual binary data sets for the four motion sensors (bedroom, corridor, lounge and kitchen) are ranged between 99% and 89%.

Therefore, it is apparent that even using a data with some irregular patterns, we would still be able to predict the movement activities of an occupant living in an IIE.

7.5.1.2 Results for Converted Input Data

The sensor data after conversion from binary to start-time and stop-time method is also used as input to NARX network. The input is the converted data using the start-time and the stop-time at time t, while the output is the start-time and the stop-time at the time $t + \tau$. In this case, τ represents the next start-time and stop-time cycle. The results for both the simulated data sets and the case studies are demonstrated below:

A) Simulated Environment

The main aim of the experiments in this section are to show the effect of increasing the number of step ahead prediction on the network performance. The capability of the system to predict the occupancy behaviour was assessed by investigating the effect of increasing the number of step ahead τ in network performance convergence.

In many research publications, a great attention has been devoted to time series prediction, and the way to predict one step ahead up to time horizon (also known as leads time or predict horizon). The prediction of more than one step ahead can be achieved by using an iterative method or a direct method. In an iterative method the prediction is accomplished by doing repeated one-step-ahead predictions up to the desired horizon, whilst in direct modelling, a model is explicitly trained to predict τ-steps-ahead [4].

The experiments are carried out on the input sensor data sets. For instance, Table 7.1 summaries the prediction results of NARX network for the lounge motion sensor dataset generated from the simulated data with 6% duration uncertainty for a period of ten days. In this experiment, two hidden layers with ten neurons in each layer are used. Two inuput and output tapped delay are used. From the results in Table 7.1. RMSE is increased just after 3 steps ahead prediction for this data set. In this table, step means the next start-time and stop-time cycle of the occupant movement activity. The results show that as τ is increased for a number of steps the RMSE is also increased. The increasing amount depends on the type of the sensor, number of hidden neurons/layers and the number of input and output memory order. It can be concluded that increasing the number of step ahead prediction time results in decreasing the accuracy of the network convergence.

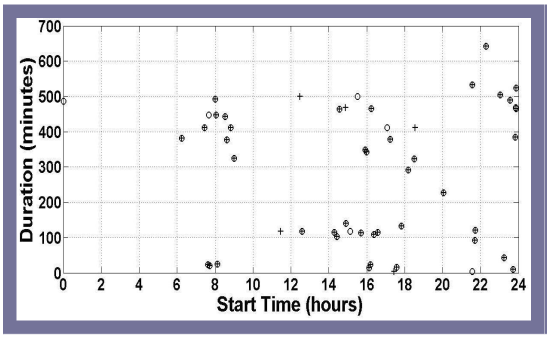
A one step ahead prediction results of the simulated data with 6% uncertainty using NARX network are shown in Figure 7.5. The input and output tapped delay D_x and D_y values are set to 6 for the input data sets. To measure the accuracy of prediction based on start-time and stop-time method of conversion, RMSE is used. The RMSE for both training and testing of the simulated data sets range from minimum of 7% to maximum of 9%.

B) Real Environment

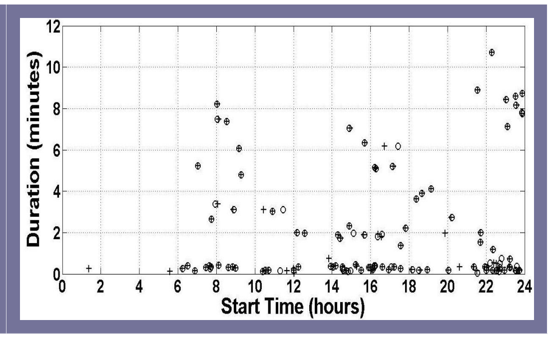
NARX network is also exploited to test the case studied collected from the real environments to monitor the movement of a person in his/her home. For instance, the data sets of case study I are trained using NARX network where input and output tapped delay are set to 10 and using 20 hidden units. RMSE to train NARX network for all real data sets are ranging from a minimum of 3% to a maximum of 9%. The prediction results of the backdoor, lounge and kitchen motion sensor data of case study I are illustrated in Figure 7.6. In these experiments, one step ahead (next start-time and stop-time cycle) is predicted. From Figure 7.6-a, it is obvious that the back door is usually opened about 1 minute or a bit more, and it is sometimes left open for more than 20 minutes.

Table 7.1: A sample of ten days training prediction results for the lounge motion sensor data with different values for number of step ahead prediction, step $*$ represents the next start-time and stop-time cycle.

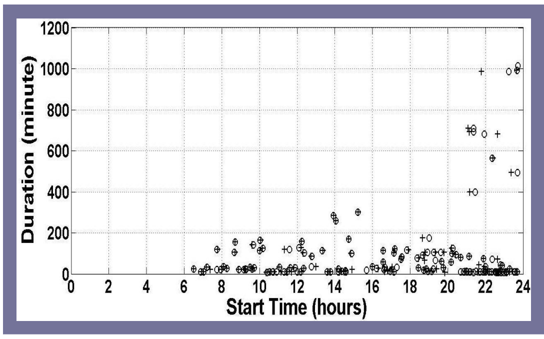
Number of step ahead	RMSE
1 step*	0.0765
2 step	0.0697
3 step	0.0977
4 step	0.5783



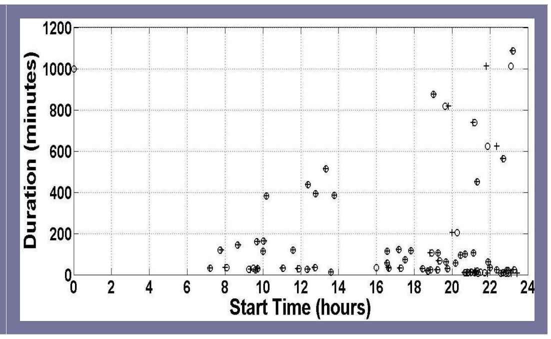
(a) Bedroom



(b) Corridor



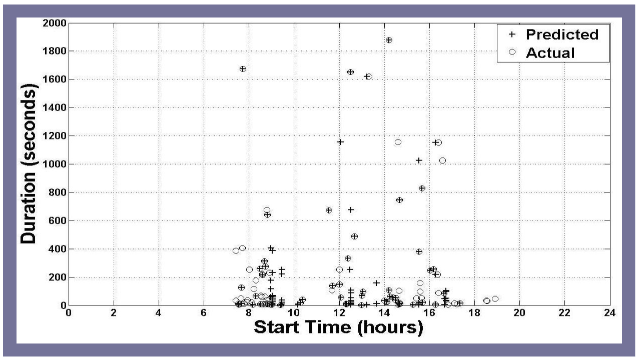
(c) Lounge



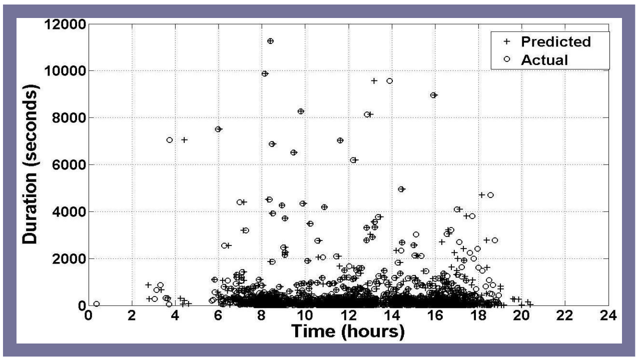
(d) Kitchen

Figure 7.5: Plots of the four sensory data sets generated from the simulated environment with 6% uncertainty based on start-time and duration time using NARX network. (+ Predicted values; O Actual values)

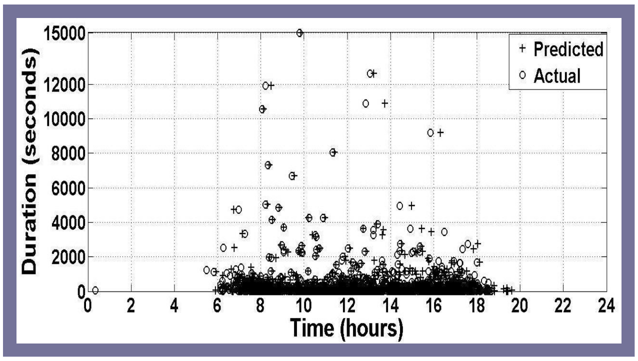
Using NARX network, experiments are also conducted on the real environment of the case study III. In these experiments, the data is split into a sample of 17 days for training (i.e. about $1,468,800$ points of binary values and 3 days for testing (i.e. about $259,200$ points of binary values). In these experiments, the network is trained using time delay $Dx = 6$ and $Dy = 6$ and two layers of hidden units (20 and 1 units respectively in each layer). For instance; Figure 7.7-a to Figure 7.7-c show the NARX prediction results of three sensors of the environment of case study III. These sensors are: bedroom motion sensor, lounge motion sensor and bath door sensor. The inputs to NARX network are the converted time series data. RMSE to train NARX network for these sensor data sets are 3%, 6% and 7% respectively. In these experiments, one step ahead (next start-time and stop-time cycle) is predicted.



(a) Back door sensor



(b) Lounge motion sensor



(c) Kitchen motion sensor

Figure 7.6: The prediction results of the three sensors of the real data collected from the environment of case study I using NARX network. (+ Predicted values; O Actual values)

Experimental results also showed that as the number of hidden neurons are increased the convergence becomes better although the training time is increased. For instance, consider training the data collected from the kitchen motion sensor on case study I where the number of input and output tapped delay are 100. In this case, the RMSE are 0.0319 for 20 hidden neurons and 0.0263 for 50 hidden neurons.

It can be concluded that, NARX network show much faster convergence in terms of accuracy of the results, and the number of epochs and training time compared with other TDNNs.

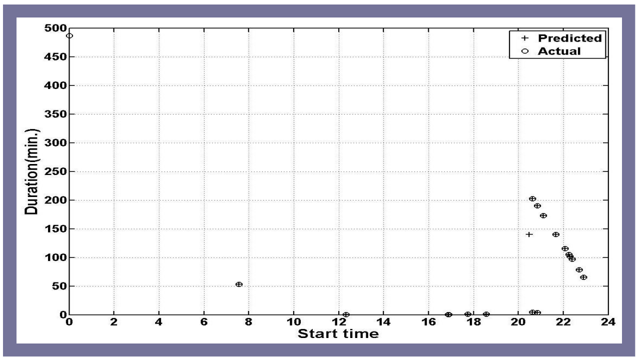
7.5.2 Prediction Results using RNNs

Experiments are also conducted using RNN techniques to predict the next movements of an occupant in an IIE. In these experiments, RNNs are successfully used for time series prediction for both the binary and converted input data. Considering the number of time series predictive models, only the results of ESN are demonstrated. The choice is made based on the high robustness and better accuracy of prediction are obtained using this network. Large number of hidden units are used to train the ESN. For example, 5000 hidden units are used to train the lounge motion sensor data set in case study I. Sigmoid function[1] is used for activation function of the hidden units and linear activation function for the output units. Sparse density of reservoir weights matrix have set to take values between 0.01 to 5.0. The spectral radius to scale reservoir weights ranges from 0.1 to 0.9. The initial values of input weights and backwards weights have intervals between $(-0.5, 0.5)$.

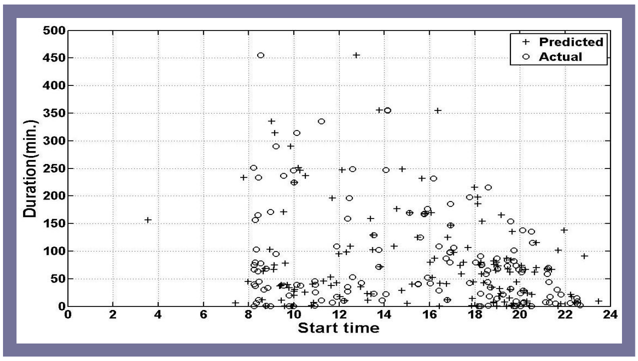
7.5.2.1 Results for Binary Input Data

As with NARX network, the predictive performance of ESN is also evaluated on the data generated from the simulated environment of an occupant with regular and less regular behaviour. For example, Figure 7.8 shows the analysis results of ESN method using 50 hidden neurons to predict two hours step ahead i.e. $\tau = 120$. The results are demonstrated for the bedroom, corridor, lounge and kitchen motion sensors data sets. Shown results are only for a sample of three days data set of a total of ten days split into training and testing data. The bedroom sensor has shown a very good match between the predicted and actual

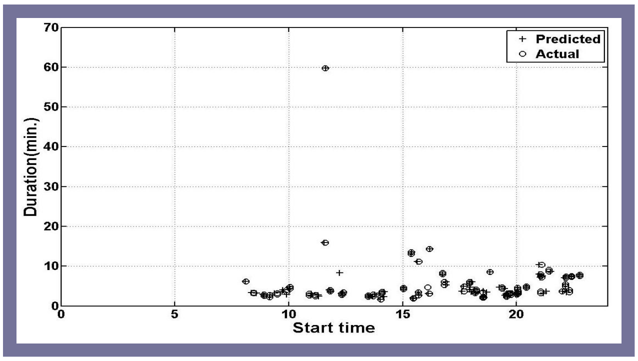
[1] $y = \frac{1}{1+e^{-x}}$



(a) Bedroom motion sensor



(b) Lounge motion sensor



(c) Bath door entry sensor

Figure 7.7: The prediction results of the three sensors of the case study III of the real data using NARX network. (+ Predicted values; O Actual values)

sensor values. The prediction performance accuracy of ESN to compare between the predicted and actual binary data are done using similarity measurements. Figure 7.9 shows the similarities between these two binary data sets generated from the simulated occupancy signal of Figure 7.3.

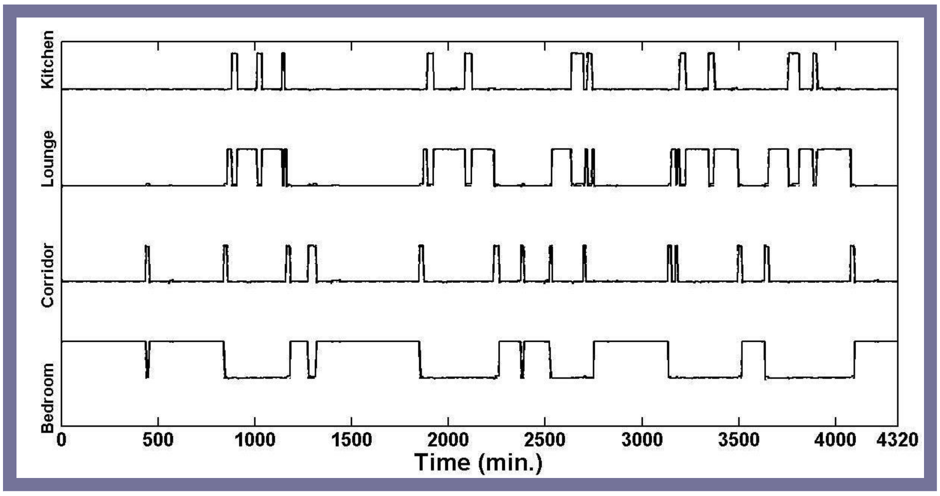


Figure 7.8: Sensor values and two hours ahead predicted values using ESN network for three days sample of simulated activities data for less regular occupant (6% duration uncertainty), (-Predicted values; − − − Actual values).

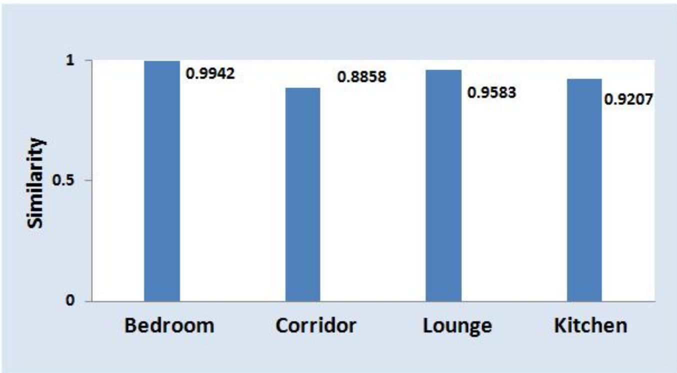


Figure 7.9: Similarity measurement accuracy between the actual and the predicted binary data generated from the simulated occupancy signal of Figure 7.8 using ESN.

Different sizes of reservoir (number of hidden neurons) are also used to test the performance of ESN. In Figure 7.10 the training time using different reservoir sizes (hidden neurons) are shown. The figure clearly shows that ESN is relatively fast and data sets are trained in only a few minutes or even seconds.

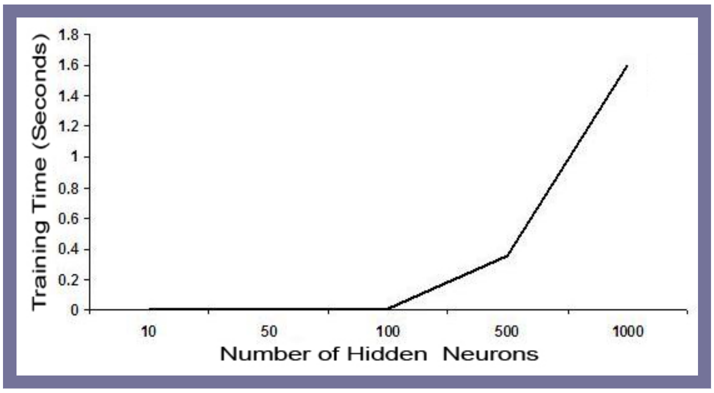


Figure 7.10: ESN training time for different reservoir sizes.

7.5.2.2 Results for Converted Input Data

The results for the converted input data for both the simulated data sets and the real case studies using ESN are demonstrated below:

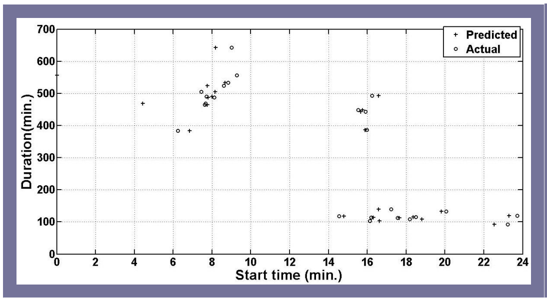
A) Simulated Environment

In this section, experiments on the converted data sets generated from the simulated environment are presented. The prediction results of the simulated data with 6% uncertainty using ESN network is shown in Figure 7.11. One step ahead prediction is made in these experimentes. The input data sets include: bedroom, corridor, lounge and kitchen motion sensors. The RMSE for testing the four sensors data set generated from the simulated environment range from a minimum of 3% to a maximum of 7%. The experiments are conducted using different ESN reseviour initialization to test the accuarcy of ESN. The resulting performance of ESN are not affected a lot.

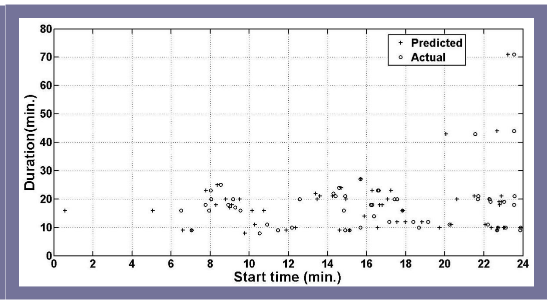
B) Real Environment

Experiments using ESN are also conducted on the real environments. The main idea of these experiments are to show the effectiveness of ESN in learning the temporal relationships between sensor data sets and predicting the time series data collected from such sensors.

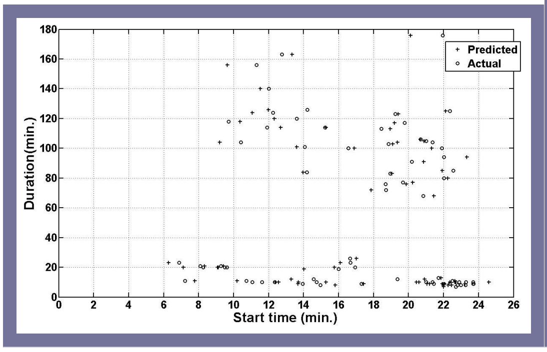
Using the data representation method discussed in Section 5.2, the sensor data sets of case study I are clustered. From the clusters illustrated in the previous



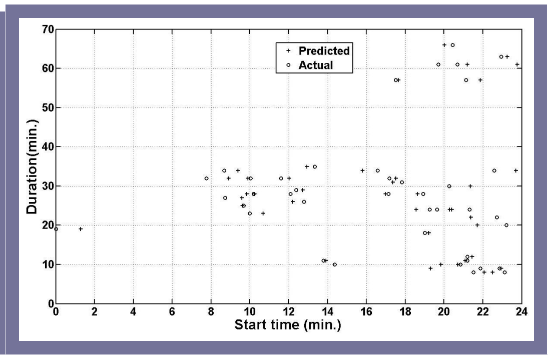
(a) Bedroom



(b) Corridor



(c) Lounge

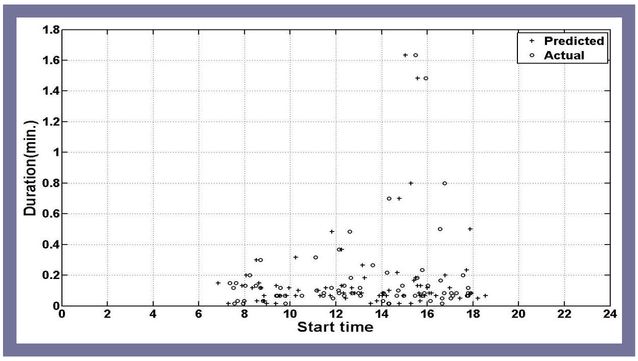


(d) Kitchen

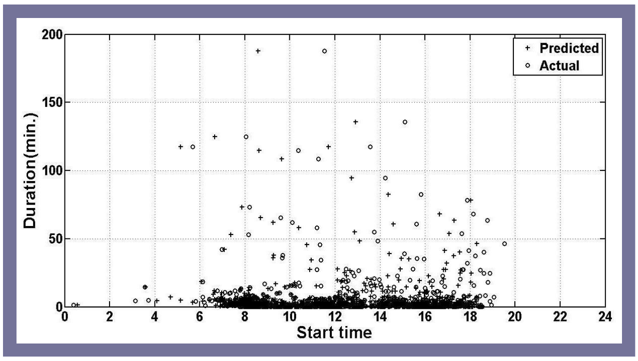
Figure 7.11: Plots of the four sensory data sets generated from the simulated environment with 6% uncertainty based on start-time and duration time using using ESN. (+ Predicted values; O Actual values)

chapter (see Figure 6.3-a and Figure 6.3-b) for front door and back door sensors data sets respectively, it is evident that for most days the front and back doors were opened for a short period of time. However, in some instances both doors were left open for a long period of time. Using this form of data representation and visualisation, we have managed to look at each sensor activity independently and identify the abnormal behaviour for that specific activity. However, all these activities are interdependent and we should be able to establish the dependency between the activities. To achieve this, an ESN is used to learn the temporal relationship of the sensors data sets in case study I environment.

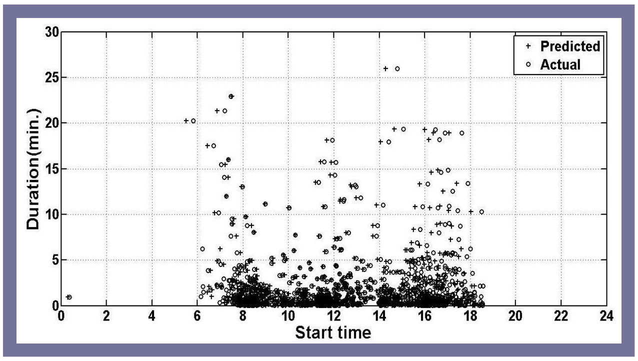
Figures 7.12 show the ESN training results of case study I for the back door, lounge motion sensor and kitchen motion sensor respectively. The prediction is based on 14 days for training (i.e. about 1,209,600 points of binary data sets)



(a) back door sensor



(b) lounge sensor



(c) kitchen sensor

Figure 7.12: The prediction results of the three sensors of case study I of the real data using ESN. (+ Predicted values; O Actual values)

and 6 days for testing (i.e. about 518, 400 points of binary time series). The experimental results are based on using 0.3 for the sparse density of reservoir weights matrix and 0.9 for the spectral radius. The number of hidden neurons for each data sets are varied form 3000 to 7000 units. and $\tau = 1$ step ahead prediction are used. The results shown here are the prediction for the 20 days. The RMSE for testing all the sensor data sets range from a minimum of 4% to a maximum of 9%.

The effects of using different number of hidden units is also investigated. Table 7.2 illustrates a comparison between the actual values with the predicted values for the lounge motion sensor data set using different number of hidden units. The table contains a sample of 20 days of data sets in case study III. The results are based on start-time and stop-time conversion. The table clearly shows that as the number of hidden neuron are increased, the network convergence is improved. It should be noted that the prediction results of the converted time series data requires large number of hidden units compared with the actual binary time series data.

7.6 Prediction Results Verification

To verify the predictive models and to see how accurate these models, a standard bench mark set of data is used. The data is collected from a two bedroom smart home in Washington State University (WSU) campus [34]. The home is occupied by a volunteer women adult. She has got visitors on a regular basis from her children and grand children. The smart home testbed consists of two bedrooms, a living dining room, a kitchen, an office room and two bathrooms. The layout of this home is illustrated in Figure 7.13. As shown in this figure, different sensors are distributed in the home. However, only motion and door entry sensors are used in this research. The sensors events that are generated

Table 7.2: Prediction results of the lounge room motion sensor data set of case study III using ESN.

No. of hidden neurons	Training RMSE	Testing RMSE
50	0.1196	0.1212
1000	0.0914	0.1001
3000	0.0771	0.0883
5000	0.0751	0.0879
7000	0.0719	0.0863

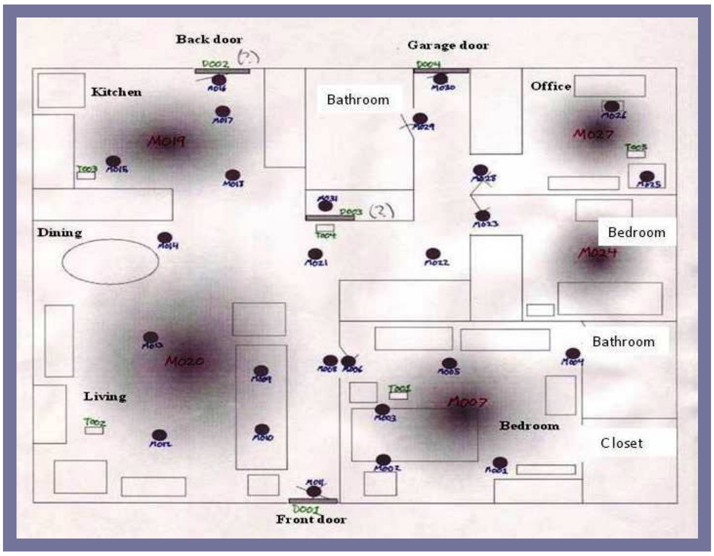
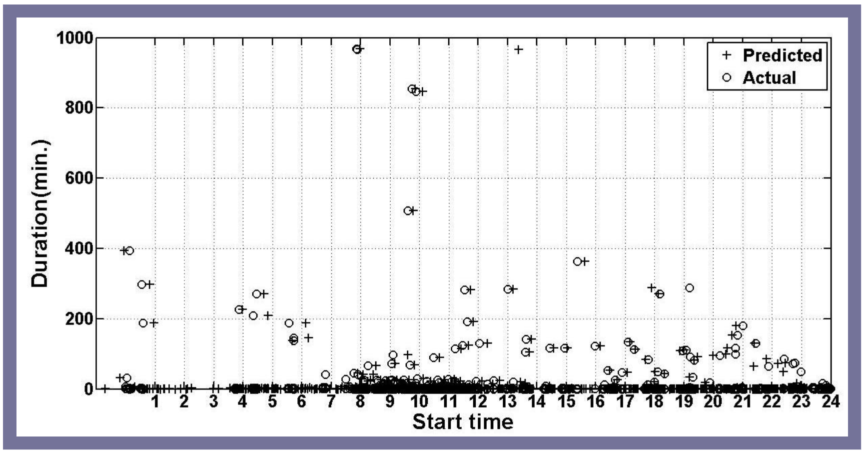


Figure 7.13: The layout of the WSU CASAS smart home [34].

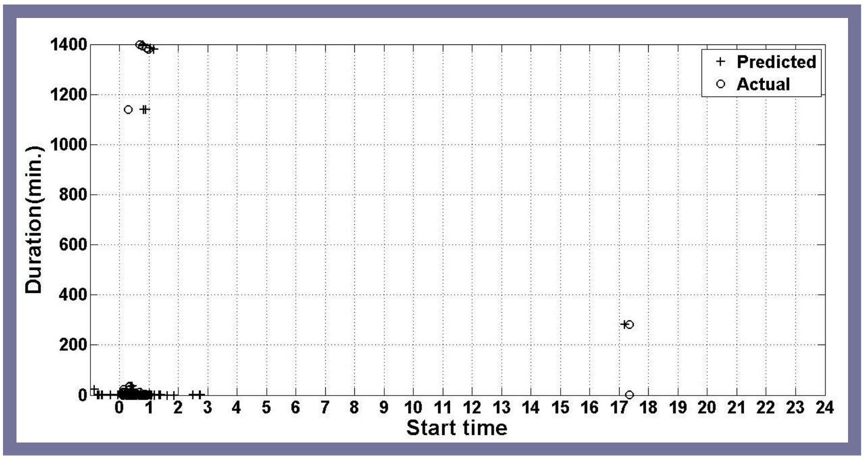
from motion sensors are started with IDs ”M”’ and door entry sensors with IDs ”D”. Different activities are carried out by the occupant of this environment including: preparing meal, sleeping, eating, working etc. The data is recorded from November 2010 to June 2011.

The results reported in this section are based on the ESN and NARX networks using the WSU CASAS data set. Different experiments are conducted on this data. In these experiments, the data is split into a sample 7 days for training and 3 days for testing. In addition, one step ahead (next start-time and stop-time cycle) is predicted in these experiments. For NARX network, the input and output tapped delay are set to 2 and using different hidden units. The inputs to these networks are the sensor data after conversion from binary to start-time and stop-time method at time t. The output is the start-time and the stop-time at the time $t + \tau$. The prediction results of two sensors data collected from the CASAS environment for both ESN and NARX networks are shown in Figure 7.14 and Figure 7.15. These sensors are bathroom motion sensor and back door entry sensor.

To measure the accuracy of prediction based on the start-time and stop-time method of conversion using ESN and NARX network, RMSE is used. For instance; Table 7.3 depicts the performance of the ESN and NARX networks in



(a) Bathroom motion sensor

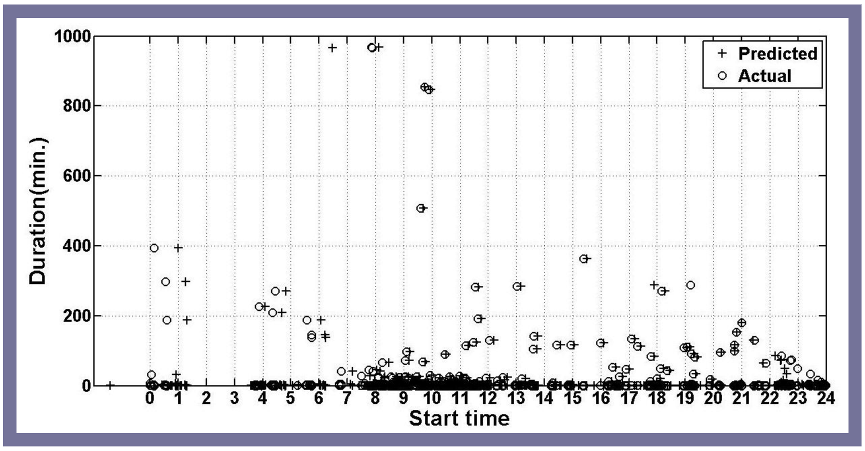


(b) Back door sensor

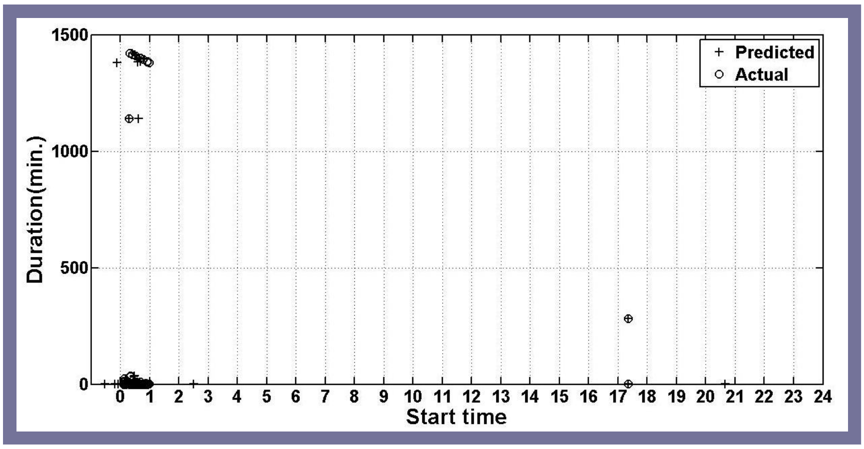
Figure 7.14: The prediction results of the two sensors of the environment (layout in Figure 7.13) of the real data using NARX. (+ Predicted values; O Actual values)

terms of RMSE training and testing using bathroom motion sensor and back door entry sensor. The performance of ESN and NARX network are compared using different number of hidden neurons and epochs. It has been observed, for certain sensor data sets, there is an increase in the prediction accuracy when ESN uses a large number of hidden neurons compared with NARX network. On the other hand, the NARX network needs many epochs to achieve high accuracy.

From this numerical analysis, comparing the ESN and NARX network results with the previous numerical results for the three case studies, which are used



(a) Bathroom motion sensor



(b) Back door sensor

Figure 7.15: The prediction results of the two sensors of the environment (layout in Figure 7.13) of the real data using NARX. (+ Predicted values; O Actual values)

in this thesis, ESN and NARX networks give good prediction results for all data sets. Therefore, it has been demnstrated that these networks are able to represent the dynamical behaviour of a system more efficiently in general applications.

7.7 Neural Networks Performance Comparison

The neural network techniques have been compared to show their ability in a single and multiple step ahead prediction of binary time series. The outputs of these techniques have shown a reasonable accuracy. The comparison depends on several factors that determine the efficiency of these techniques. The computation time which are required for training are the most important. Experiments are contacted on the simulated data and the real data. The predictive model are compared the TDNNs and RNNs to predict the binary time series data collected from sensor network. These networks are Elman network trained using BP and the BPTT algorithms, echo state network, NARX network. The networks are trained using the same parameters in these experiments (i.e. the number of hidden layers and units etc.) are the same for all networks. For example, Table 7.4 compares the results of all sensor datasets using the neural network techniques for time series prediction. The prediction is based on the start-time and stop-time as inputs to the networks.

In these experiment, the number of the hidden units play an important role in network convergence. Increasing the number of the hidden units results in decreasing the RMSE of training for both networks. However, the training time and number of epochs that are required to train the network increases rapidly in Elman network in contrast to ESN and NARX. For instance, training the lounge sensor data using Elman network requires 1000 epochs for training, while only 50 epochs are required using NARX network. It can be concluded that the ESN and NARX network prediction results have the best results compared with Elman network trained with either BP or BPTT in that the training time is significantly shorter. The other approaches suffer from slow convergence as the number of neurons are increased.

The RNNs are also compared with the TDNNs using the real data. Based

Table 7.3: Prediction results of the CASAS smart home data set using ESN and NARX networks. ($\oplus$: Sensor ID ; $\otimes$: No. of hidden neurons; $\ominus$: Training RMSE; and $\oslash$: Testing RMSE.)

Method	$\oplus$	$\otimes$	$\ominus$	$\oslash$
ESN	Bathroom motion sensor	40	0.0690	0.0681
	Back door entry sensor	42	0.0898	0.0833
NARX network	Bathroom motion sensor	40	0.0631	0.0641
	Back door entry sensor	42	0.0072	0.0979

on our experiments, the convergence of NARX network is much better than the Elman network to train the real data. For example, the RMSE for training the back-door and front-door sensor data using Elman network are 1.75 and 2.15 respectively. However, the RMSE is only 0.05 and 0.07 to train these door entry sensors using NARX network.

To summarize, the predictive algorithms can be classified as a way of comparing based on the accuracy of the algorithm and the convergence time. All algorithms are classified in four groups A, B, C and D as shown in Table 7.5. The letter A represents a maximum, while the letter D represents a minimum. As shown in the above table, it was found the RNN trained with ESN and LSTM has the highest accuracy and the lowest convergence time.

Table 7.4: Prediction results of all sensor datasets using ESN, Elman network, FTDD, NARX. ($\oplus$: No. of hidden neurons; $\otimes$: Training RMSE; $\ominus$: Testing RMSE; and $\oslash$: Time(Sec.).)

Method	$\oplus$	$\otimes$	$\ominus$	$\oslash$
ESN	10	0.0556	0.0556	0.0116
ESN	50	0.0556	0.0556	0.0519
Elman network trained using BP	10	0.2027	0.2964	181.2131
Elman network trained using BP	50	0.1031	0.1740	241.0627
Elman network trained using BPTT	10	0.0759	0.0766	12.7315
Elman network trained using BPTT	50	0.0803	0.0811	218.0986
NARX network	10	0.0738	0.0740	16
NARX network	50	0.0729	0.0731	1.46

Table 7.5: The classification of the algorithms with respect to the accuracy of the algorithm and the convergence time.

Algorithm	Accuracy	Time
Focused Time Delay	D	B
Layer Recurrent Network	B	D
NARX	A	B
Echo State Network	A	A
Long Short Term Memory	A	A
Elman network trained using BPTT	D	A
Elman network trained using Real Time Recurrent Network	D	C
Recursive Self-Organizing Map	C	D

It can also be concluded that TDNNs representing by LRN and FTDNN networks have similar performance as the network trained by NARX, but it usually required much more training time and more than one thousand of training epochs. A common feature of these approaches is that when the number of epoch and the number of hidden layers are increased an improvement on the accuracy of the error is seen.

For RNN techniques representing by Elman networks trained with BP or BPTT learning algorithms, their predictive performance are low compared with ESN and LSTM. Elman network trained with these algorithms are usually influenced by initial weights and can not converge to global minima. For convergence, these networks need more hidden layers, hidden units, and training time and epochs.

7.8 Discussions

This chapter examined the use of several neural networks to predict the behavioural patterns of an occupant living in an IIE. A comparison between these networks is also made. The results of this study indicate that, in general, these networks produce an accurate prediction. However, ESN, LSTM and NARX network, compared to the other predictive techniques yield the highest accuracy in prediction.

It should be noted that the type of the input data set has an impact on the prediction capability of the models. The predictive neural network models are applied for both binary and converted data sets such as start-time and stop-time representation format). Although the predictive models give good results in predicting the binary inputs, the predictive models use the compressed data as inputs instead of the large sparse binary data sets. The amount of memory and the number of computation used are reduced for the converted input data. Also, in contrast to the binary input data, the prediction results of the converted data to predict a single step ahead yields the prediction of the next inhabitant's movement (next start-time and stop-time cycle).

Concerning the convergence of TDNNs, more than one hidden layer is needed to converge when using the converted input data. For instance; consider training the data collected from the kitchen motion sensor of the real environment of case study I using NARX network. In these experiments, number of input and output tapped delay Dx and Dy are 100. The RMSE is 0.0919 using two hidden layers, while the RMSE is 1.24 for the network with one hidden layer. For RNNs such as ESN need only one hidden layer with a large number of reservoir (hidden units) is required to predict the converted input data.

Based on the prediction results from using ANNs, a classification of these networks shows that for TDNN, the NARX network gives good results in predicting the binary time series data collected from an intelligent environment. The ESN and LSTM for RNNs perform better than the simple RNNS such as Elman network. For instance, ESN is a very good choice for time series data prediction generated from the sensory data because in the methodology of sensor networks, new data is arriving at any time, whilst other approaches need all data input at the same time steps in order to compute the output.

The experimental results were successful as the network was able to successfully recognise the behavioural patterns of an occupant.

Chapter 8

Conclusions and Future Works

8.1 Summary

The work presented in this thesis is a novel attempt to answer the research question both from the practical and theoretical point of view. Based on the results obtained from this research, it can be concluded that a home equipped with some low-level sensory devices can provide important information about the status of the occupant. The proposed approach works better for elderly residents when more routine activities are expected. Specifically, we focused our research for elderly patients who are suffering from dementia.

The aim of the research was to investigate efficient mining of useful information from a sensor network forming an ambient intelligence environment. We have investigated methods for supporting independent living of the elderly by means of equipping their home with a simple sensor network to monitor their behaviour and identify their ADL. The research was conducted to enhance the efforts for better understanding, behavioural identification and predication within AmI environments.

Data provided for our investigation were collected from real environments as well as a previously developed simulator. The simulator was modified to include trending behaviour in the ADL. For real data, reported as three different case studies, door entry and occupancy sensors were used to extract the movement patterns of the occupant. Sensors produced long sequences of binary data and different approaches for data visualisation and compression were investigated. Different techniques were also investigated to identify outliers or abnormal behaviours in ADLs. ANN and specifically RNN are used to compare their abilities in accurately predicting the behavioural patterns of an occupant.

In summary, throughout this research, original knowledge on binary data visualisation, identification and prediction has been obtained. In the remaining part of this chapter, the research conclusions with critical discussion and direction of some future works are presented.

8.2 Concluding Remarks

This thesis attempts to provide an analysis of identification and prediction of abnormal behaviour activities of daily living in IIEs. Conclusions for different aspects of the project are presented below.

8.2.1 Activity Representation and Visualization

This thesis highlights the need for a flexible and efficient data representation and visualisation techniques in large binary sensor data sets. Data representation techniques are used and evaluated using the simulated and the real environments. For example, the start-time and stop-time approach is successfully used to convert and represent the binary sensors data sets. In addition, by using the starts-time and duration, we have been able to clearly visualise large data sets, and detect abnormalities in the occupant behaviour. The results of these conversion techniques produced compressed binary data compared with the original long series of binary data items. Moreover, various clustering and visualization techniques such as fuzzy c-mean and PCA are used to visualize and find anomalies and outliers within sensor data.

8.2.2 Abnormal Activity Identification

In this thesis the relationships between frequent patterns of user activities representing the ADL in an IIE are also studied. We were interested in identifying similarities and the differences between everyday activities. As a result of this research, a good understanding of the behaviour of occupants are identified. In three real case studies reported here, occupancy sensor datasets are collected from elderly people.

This study has particularly investigated the effectiveness of the binary (dis) similarity or distance measures in an assisted living environment. In this investigation, our aim was to assess these measures to identify the similar and abnormal behaviour patterns of an occupant in an intelligent environment. Different indices were applied in finding the mismatching bits between two binary vectors representing two different activities. The results of this investigation show that fuzzy hamming distance is better than other distance measures for binary data generated from motion sensors since these kinds of sensors are normally having significant changes. Fuzzy hamming distance gives less distance than the classical hamming distance for such sensors. However, as the dimension of the distance measures is increased, it is rather difficult to identify the anomalies and outliers within the sensor data.

To tackle the high dimensionality of the data and to identify the outliers and anomalies of user's activities behavioural patterns, a two stages outlier detection is proposed integrating PCA and FRBS. The principal components which are computed using PCA, are used to identify two indices the residual SPE and Hotelling's T^2. Residual SPE and Hotelling's T^2 is used to improve the results of the PCA. Eventually the outliers and abnormal behaviour of an inhabitant are identified using FRBS. The proposed system is successfully able to classify the normal and abnormal behaviour or outliers. The severity of the outliers are found as well. The FRBS is designed to generate alarms when any abnormality is detected for elderly people who are lived independently. The relationships between similar patterns are summarized in a meaningful manner by regularly generating a symbolic report in natural language based on the extracted patterns.

The proposed techniques are evaluated by applying to the simulated environment as well as real environments to identify and predict the occupancy movement behaviour patterns using computational intelligent techniques. Additionally, the results and knowledge gained from of this research are not limited to the field of intelligent environments only, and it could be also used in other domains such as wildlife monitoring and traffic monitoring.

8.2.3 Activity Extraction and Prediction

This thesis also shows that the occupancy pattern extraction and prediction in an intelligent environment can be efficiently modelled using ANNs. Different ANN techniques were applied to binary sensor time series prediction. These techniques can predict the next movement of the occupant, and can also give us the duration that the occupant stays in a specific area in the environment. Sensors were used to record the behaviour of the occupant and allow the carer to observe any changes to patterns. The predicted results from these networks were tested and compared to show their ability in a single and multiple step ahead prediction.

In general, the major findings of this work in terms of activity prediction of binary time series data, collected from occupancy sensors in an IIE, are listed below:

- Very limited research are reported in literature on analysis and prediction of binary time series. Specifically this was true for analysis of any binary time series representing the data collected from intelligent environments. This research has helped with better understanding and investigating prediction techniques for binary time series.

- Large volume of data collected from sensor data are used to predict the next movement activities of an occupant. The raw data collected every second could lead to a large space data set. For example in a real environment the data collected from only one occupancy sensor could be over 2.5 million entry over a period of one month. For any analysis and prediction, it is essential to compress the data without losing any important features of the original data set.

- Using start-time and stop-time form of compression has helped to be able to predict the next activity rather than τ step ahead prediction.

- The demonstrated results indicate that temporal neural network algorithms outperformed statistical and traditional time series prediction methods. For TDNNs, NARX network gives better results than simple recurrent networks, due to the feedback from the output layer to the input layer using a proper number of input time delays. Additionally, the results presented in this research show that ESN and LSTM are very promising approach for binary datasets collected from smart environments.

- The importance of feature extraction from the raw data (time sequence) as inputs to the predictive models is investigated.

- The impact of different parameters of recurrent neural network algorithms is explored. Examples of such parameters are: the number of hidden layers, hidden units, number of step ahead, input/output memory order, training time and maximum adopted time delay.

Overall, this research could play a vital role in the fields of IIEs with more development.

8.3 Directions for Future Works

Further investigation, in which future works could proceed, are listed below:

- A further direction for investigation is to implement the approaches presented in this thesis for a multiple occupancy situation, i.e. more than one occupant in a smart flat/home. The approaches would not be efficient in presence of visitors or even when the elderly people have a pet companionship which is true for some cases. This task is complex, as not only must monitor the occupant, but it must also predict whether he/she alone or has visitors in a specific time.

- This study will serve as a base for future studies in which a combination of discrete sensors (occupancy, door entry point, ...) and continuous sensors (temperature, humidity, ..) will be used. The current study has only examined the binary similarity and distance measures. A future study investigating the impact of using sensor data from other sensors (e.g. non-binary) that may be found in smart homes to provide a larger data sample for the experiments would be very interesting. In addition, it would be better to use sensor data that can handle multi-attribute vectors. Further work also needs to be done to assess the weighted binary measures on a data collected from occupancy sensors. The matches and mismatched bits are weighted according to their information as obtained in the actual sequence values.

- It would be interesting to build a friendly user interface system to allow the end user of the activity prediction algorithms to train and test the system on-line.

- It will be interesting to extend the work to develop the semantic modelling of the behaviour of an occupant where the predicted values are communicated with the elderly and carer in linguistic terms. This will be achieved by taking the results of this thesis and feeding them to a software named Protégé [142]. Protégé is an open source ontology editor and knowledge base framework is used.

- The predictive techniques used in this work could be extended to predict more complex human behaviour. For future planning, some predictive algorithms would be investigated to recognize more complex human behaviour activity.

Appendix A

Prediction Algorithm Using a HMM of Order 1 [59]

1. $T = 1$ (T is the length of the observation sequence);

2. $T = T + 1$;
   if $T < H$ go to 2.

3. c=0 (c is the number of current iteration, its maximum value is given by I);

4. The model $\lambda = (A, B, \pi)$ is repeatedly adjusted based on the last H observations $O_{T-H+1}, O_{T-H+2} \dots O_T$ (the entire observation sequence if $H = T$), in order to increase the probability of the observation sequence $P(O_{T-H+1}O_{T-H+2} \dots O_T|\lambda)$. In a, b and c steps the denominators are used in order to obtain a probability measure, and to avoid underflow. Underflow is inevitable without scaling, since the probabilities tend to 0 exponentially as T increases.

   (a) Compute the forward variable α in a recursive manner:
   $\alpha_{T-H+1}(i) = \frac{\pi_i . b_i(O_{T-H+1})}{\sum_{i=0}^{N-1} \pi_i . b_i(O_{T-H+1})}, i = 0, \dots, N-1$, where $\alpha_{T-H+1}(i)$ is the probability of observation symbol O_{T-H+1} and initial hidden state S_i, given the model $\lambda = (A, B, \pi)$;

   (b) $\alpha_t(j) = \frac{\sum_{i=0}^{N-1} \alpha_{t-1}(i).a_{ij}.b_j(O_t)}{\sum_{j=0}^{N-1}\sum_{i=0}^{N-1} \alpha_{t-1}(i).a_{ij}.b_j(O_t)}$, where $t = T - H + 2, \dots, T, j = 0, \dots, N-1$ where $\alpha_t(j)$ is the probability of the partial observation sequence until time $t(O_{T-H+1} \dots O_t)$, and hidden state S_j at time t, given the model $\lambda = (A, B, \pi)$. Since, by definition,
   $\alpha_T(j) = P(O_{T-H+1}O_{T-H+2} \dots O_T, q_T = S_j|\lambda)$,
   the sum of the terminal forward variables $\alpha_t(j)$ gives the probability

of the observation sequence:
$P(O_{T-H+1}O_{T-H+2}\dots O_T|\lambda) = \sum_{j=0}^{N-1}\alpha_T(j)$.

(c) Compute the backward variable β in a recursive manner:
$\beta_T(i) = \frac{1}{\sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\alpha_{T-1}(i).a_{ij}.b_j(O_T)}, i = 0,\dots,N-1;$
$\beta_t(i) = \frac{\sum_{j=0}^{N-1}a_{ij}.b_j(O_{t+1}).\beta_{t+1}(j)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}a_{ij}.b_j(O_{t+1}).\beta_{t+1}(j)}, t = T-1,\dots,T-H+1, i = 0,\dots,N-1;$
where $\beta_t(i)$ is the probability of the partial observation sequence from $t+1$ to the end T $(O_{t+1}O_{t+2}\dots O_T)$, given hidden state S_i at time t and the model $\lambda = (A, B, \pi)$.

(d) Compute ξ:
$\xi_t(i,j) = \frac{\alpha_t(i).a_{ij}.b_j(O_{t+1}).\beta_{t+1}(j)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\alpha_t(i).a_{ij}.b_j(O_{t+1}).\beta_{t+1}(j)}, t = T-H+1,\dots,T-1, i = 0,\dots,N-1, j = 0,\dots,N-1;$
where $\xi_-(i,j)$ is the probability of being in hidden state S_i at time t and respectively S_j at time $t+1$, given the observation sequence $O_{T-H+1}O_{T-H+2}\dots O_T$ and the model $\lambda = (A, B, \pi)$.

(e) Compute γ:
$\gamma_t(i) = \sum_{j=0}^{N-1}\xi_t(i,j), t = T-H+1,\dots,T-1, i = 0,\dots,N-1;,$
where $\gamma_t(i)$ is the probability of being in the hidden state S_i at time t, given the model $\lambda = (A, B, \pi)$ and the observation sequence $O_{T-H+1}O_{T-H+2}\dots O_T$.

(f) Adjust π:
$\overline{\pi} = \gamma_{T-H+1}(i)$-represents the expected number of times the hidden state is S_i at the initial time $t = T-H+1$.

(g) Adjust A:
$\overline{a_{ij}} = \frac{\sum_{t=T-H+1}^{T-1}\xi_t(i,j)}{\sum_{t=T-H+1}^{T-1}\gamma_t(i)}$ -represents the probability of transition from hidden state S_i to S_j. The numerator is the expected number of transitions from state S_i to S_j, while the denominator is the expected number of transitions from state S_i to any state.

(h) Adjust B:
$\overline{b_j(k)} = \frac{\sum_{t=T-H+1}^{T-1}{}_{O_t=V_k}\gamma_t(j)}{\sum_{t=T-H+1}^{T-1}\gamma_t(i)}$ -the probability of observation symbol V_k given that the model is in hidden state S_j. The numerator is the expected number of times the model is in hidden state S_j and the ob-

servation symbol is V_k, while the denominator is the expected number of times the model is in hidden state S_j.

(i) $c = c + 1$;
if $log[P(O_{T-H+1} \dots O_T|\overline{\lambda}] > log[P(O_{T-H+1} \dots O_T|\lambda]$and $c < I$ then go to 4.). Since P would be out of the dynamic range of the machine, we compute the log of P, using the following formula:
$log[P(O_{T-H+1} \dots O_T|\overline{\lambda}] = -log(\frac{1}{\sum_{i=0}^{N-1} \pi.b_i(O_{T-H+1})})-$
$\sum_{t=T-H+2}^{T} log(\frac{1}{\sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \alpha_{t-1}(i).aij.b_j(O_t)})$

5. At current time T, it is predicted the next observation symbol O_{T+1}, using the adjusted model $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$:

   – choose hidden state S_i at time T, $i = 0, \dots, N-1$, maximizing $\alpha_T(i)$;

   – choose next hidden state S_j (at time $T+1$), $j = 0, \dots, N-1$, maximizing $\overline{a_{ij}}$;

   – predict next symbol V_k (at time $T+1$), $k = 0, \dots, M-1$, maximizing $\overline{b_j(k)}$.

   If the process continues, then $T = T + 1$ and go to 3.

Appendix B

The following is the pseudo code of Back Propagation Through Time(BPTT) algorithm [87]:-

Input: current weights ω training time series

Output: New weights

Computation steps

1. Forward pass: as described in section 3.4

2. Compute , by proceeding backward through $n = T, ..., 1$ for each time n and unit activation $x_i(n), y_j(n)$ the error propagation term $\delta_i(n)$

$$\delta_j(T) = (d_j(T) - y_j(T))\frac{\partial f(u)}{\partial u}|_{u=z_j(T)} \tag{1}$$

for the output units of time layer T and

$$\delta_i(T) = [\sum_{j=1}^{T} \delta_j(T)\omega_{ji}^{out}]\frac{\partial f(u)}{\partial u}|_{u=z_j(n)} \tag{2}$$

for the output units of time layer T and

$$\delta_i(n) = [(d_j(n) - y_j(n)) + \sum_{i=1}^{N} \delta_i(n+1)\omega_{ji}^{back}]\frac{\partial f(u)}{\partial u}|_{u=z_j(n)} \tag{3}$$

for the output units of earlier layers, and

$$\delta_i(n) = [\sum_{j=1}^{N} \delta_j(n+1)\omega_{ji} + \sum_{j=1}^{L} \delta_j(n)\omega_{ji}^{out}] \frac{\partial f(u)}{\partial u}|_{u=z_j(n)} \quad (4)$$

for internal units $x_i(n)$ at earlier times,where $z_i(n)$ again is the potential of the corresponding unit Adjust the connection weights according to:-

$$new \quad \omega_{ij} = \omega_{ij} + \gamma \sum_{n=1}^{T} \delta_i(n) x_j(n-1) [use \quad x_j(n-1) = 0 \quad for \quad n = 1] \quad (5)$$

$$new \quad \omega_{ij}^{in} = \omega_{ij}^{in} + \gamma \sum_{n=1}^{T} \delta_i(n) u_j(n) \quad (6)$$

$$new \quad \omega_{ij}^{out} = \omega_{ij}^{out} + \gamma \times \{\sum_{n=1}^{T} \delta_i(n) u_j(n), if \quad j \quad refers\, to\, input\, unit \quad (7)$$

$$new \quad \omega_{ij}^{back} = \omega_{ij}^{back} + \gamma \sum_{n=1}^{T} \delta_i(n) y_j(n-1), [use\, y_j(n-1) = 0\, for \quad n = 1] \quad (8)$$

Appendix C

The following is the pseudo code of Long Short Term Memory (LSTM) recurrent training algorithm [151]:-

Forward Pass

Reset all activations to 0 Running forwards from time τ_0 to time τ_1, feed in the inputs and update the activations. Store all hidden layer and output activations at every time step. For each LSTM block, the activations are updated as follows:

Input Gates

$$x_i = \sum_{j \in N} w_{ij} y_j(\tau - 1) + \sum_{c \in C} w_{lc} S_c(\tau - 1) \tag{9}$$

$$y_i = f(x_i) \tag{10}$$

Forget Gates:

$$x_\phi = \sum_{j \in N} w_{\phi j} y_j(\tau - 1) + \sum_{c \in C} w_{\phi c} S_c(\tau - 1) \tag{11}$$

$$y_\phi = f(x_\phi) \tag{12}$$

Cells:

$$\forall c \in C, x_c = \sum_{j \in N} w_{cj} y_j(\tau - 1) \tag{13}$$

$$S_c = y_\phi s_c(\tau - 1) + y_i g(x_c) \tag{14}$$

Output Gates:

$$x_\omega = \sum_{j \in N} w_{\omega j} y_j(\tau - 1) + \sum_{c \in C} w_{\omega c} S_c(\tau) \tag{15}$$

$$y_\omega = f(x_\omega) \tag{16}$$

Cell Outputs:

$$\forall c \in C, y_c = y_\omega h(s_c) \tag{17}$$

Backward Pass

Reset all partial derivatives to 0 Starting at time τ_1 propagate the output errors backwards through the unfolds net, using the standard BPTT equations:

$$\delta_k(\tau) = \frac{\partial E(\tau)}{\partial x_k} \tag{18}$$

$$e_k(\tau) = y_k(\tau) - t_k(\tau) k \in outputunits \tag{19}$$

$$\epsilon_k(\tau_1) = e_k(\tau_1) \tag{20}$$

$$\epsilon_k(\tau - 1) = e_k(\tau - 1) + \sum_{j \in N} w_{jk} \delta_j(\tau) \tag{21}$$

For each LSTM block the delta's are calculated as follows:
Cell Outputs:

$$\forall c \in C, \epsilon_c = \sum_{j \in N} w_{jc} \delta_j(\tau + 1) \tag{22}$$

Output Gates:

$$\delta_\omega = f'(x_\omega) \sum_{c \in C} \epsilon_c h(S_c) \tag{23}$$

States:

$$\frac{\partial E}{\partial S_c}(\tau) = \epsilon_c y_\omega h'(y_c) + \frac{\partial E}{\partial S_c}(\tau+1) y_\phi(\tau+1) + \delta_i(\tau+1)\omega_{ic} + \delta_\phi(\tau+1)\omega_{\phi c} + \delta_\omega \omega_{\omega c} \tag{24}$$

Cells:

$$\forall c \in C, \delta_c = y_i g'(x_c)\frac{\partial E}{\partial S_c} \tag{25}$$

Forget Gates:

$$\delta_\phi = f'(x_\phi)\sum_{c\in C}\frac{\partial E}{\partial S_c}y_c(\tau-1) \tag{26}$$

Input Gates:

$$\delta_i = f'(x_i)\sum_{c\in C}\frac{\partial E}{\partial S_c}g(x_c) \tag{27}$$

Using the standard BPTT equation, accumulate the $\delta's$ to get the partial derivatives of the cumulative sequence error:

$$define E_{total}(S) = \sum_{\tau=\tau_o}^{\tau_1} E(\tau) \tag{28}$$

$$define \nabla_{ij}(S) = \frac{\partial E_{total}(S)}{\partial \omega_{ij}} \tag{29}$$

$$\Rightarrow \nabla_{ij}(S) = \sum_{\tau=\tau_o+1}^{\tau_1} \delta_j(\tau) y_j(\tau-1) \tag{30}$$

Update Weights

After the presentation of sequence S, with learning rate α and momentum m, update all weights with the standard equation for gradient descent with momentum:

$$\Delta\omega_{ij}(S) = \alpha\nabla_{ij}(S) + m\triangle\omega_{ij}(p-1) \tag{31}$$

References

[1] United nations department of economic and social affairs, http://www.un.org/esa/population/unpop.htm. 1

[2] European comission,(2010), europa public health, http://ec.europa.eu/health/ph_information/dissemination/diseases/alzheimer_en.htm, August 2010. 1, 4, 9, 54

[3] H. Abdi and L. J. Williams. Principal component analysis. *WIREs Comp Stat*, **2**:433–459, 2010. 98

[4] Mayur Agrawal and Sailesh Pati. Stock trend prediction using recurrent neural networks. 2006. 121

[5] M. Akhlaghinia, Ahmad Lotfi, Caroline Langensiepen, and Nasser Sherkat. Occupancy simulator for a single-occupant ambient intelligent environment. In *7th IEEE International Conference on Cybernetic Intelligent Systems*, number 9-10, pages 1–6, 2008. 62, 63

[6] M. Javad Akhlaghinia. Occupancy monitoring and prediction in ambient intelligent environment. *PhD. Nottingham Trent University*, 2010. xiv, 58, 61, 62

[7] M. Javad Akhlaghinia, Ahmad Lotfi, Caroline Langensiepen, and Nasser Sherkat. Occupant behaviour prediction in ambient intelligence computing environment. *International Journal of Uncertain Systems*, **2**[2]:85–100, 2008. xiii, 10, 12, 18, 19

[8] M. Javad Akhlaghinia, Ahmad Lotfi, Caroline Langensiepen, and Nasser Sherkat. Occupancy monitoring in intelligent environment

through integrated wireless localizing agents. *IEEE Symposium on Intelligent Agents*, pages 70–76, April 2009. 21

[9] MJ. Akhlaghinia, Ahmad Lotfi, and Caroline Langensiepen. Localising agents in multiple−occupant intelligent environments. In *IEEE International Conference on Fuzzy Systems (FUZZ), Barcelona, Spain*, pages 2202–2208, July 2010. 70

[10] MJ. Akhlaghinia, Ahmad Lotfi, Caroline Langensiepen, and N. Sherkat. A fuzzy predictor model for the occupancy prediction of an intelligent inhabited environment. *IEEE 16th International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 939–946, 2008. 9

[11] F. Albinali, N. Davies, and A. Friday. Structural learning of activities from sparse datasets. In *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'07)*, pages 221–228, 2007. 3, 14, 46

[12] Raza Ali, Mohamed ElHelw, Louis Atallah, Benny Lo, and Guang-Zhong Yang. Pattern mining for routine behaviour discovery in pervasive healthcare environments. In *5th International Conference on Information Technology and Application in Biomedicine, Hong Kong, China*, pages 241–244, 2008. 11

[13] Derek Anderson, Robert H. Luke, James M. Keller, Marjorie Skubic, Marilyn Rantz, and Myra Aud. Modelling human activity from voxel person using fuzzy logic. *IEEE Transactions on Fuzzy Systems*, **17**:39–49, February 2009. 27

[14] Louis Atallah and Guang-Zhong Yang. The use of pervasive sensing for behaviour profiling - a survey. *Pervasive and Mobile Computing*, **5**[5]:447–464, 2009. 14, 15

[15] Juan Carlos Augusto. Ambient intelligence: Basic concepts and applications. *Software and Data Technologies, Communications in Computer and Information Science*, **10**:16–26, 2008. 54

[16] Tracy Barger, Donald Brown1, and Majd Alwan. Health status monitoring through analysis of behavioural patterns. In *8th Congress of the Italian Association for Artificial Intelligence (AI*IA) on Ambient Intelligence*, pages 22–27. Springer-Verlag, 2003. 16

[17] David Bell and Lynn Palmer. Pattern matching using fuzzy methods. *CA Department of Health Services, Genetic Disease Branch*, pages 1–4. 90

[18] Manuele Bicego and Enrico Grossoand Edoardo Otranto. A hidden markov model approach to classify and predict the sign of financial local trends. In *Structural, Syntactic, and Statistical Pattern Recognition*, **5342** of *Lecture Notes in Computer Science*, pages 852–861. Springer Ber / Heidelberg, 2008. 34, 35

[19] R. Bon, M. Assaad, and M. Crucianu. Boosting recurrent neural networks for time series prediction. *Proceedings of the International Conference in Roanne (France)*, April 2003. 44, 51

[20] Abraham Bookstein, Vladimir A. Kulyukin, and Timo Raita. Generalized hamming distance. *Information Retrieval*, **5**:353–375, 2002. 90

[21] Fan M. Reynaud R. Bouaziz, S. and T. Maurin. Multi-sensors and environment simulator for collision avoidance applications. *Fifth IEEE International Workshop on Computer Architectures for Machine Perception*, pages 127–130, 2000. 62

[22] Salah Bouhouche, Malek Lahreche, Abdelkrim Moussaoui, and Jrgen Bast. Quality monitoring using principal component analysis and fuzzy logic application in continuous casting process. *American Journal of Applied Sciences*, **4**:637–644, 2007. 24, 101

[23] F. Moelaert B. Bergvall-Kareborn F. Meiland D. Craig R. Davies A. Reinersmann M. Hettinga A. . Andersson R. . Droes C. Nugent, M. Mulvenna and J. E. Bengtsson. Home based assistive technologies for people with mild dementia. *5th International Conference on Smart Homes and Health Telematics*, pages 63–69, 2007. 9

[24] Victor Callaghan, G. Clarke, M. Colley, Hani Hagras, J. S. Y. Chin, and F. Doctor. Inhabited intelligent environments. *BT Technology Journal*, **22**:233–247, 2004. 9

[25] R. Casas, R. Blasco Marn, A. Robinet, A. R. Delgado, A. R. Yarza, J. Mcginn, R. Picking, and V. Grout. User modelling in ambient intelligence for elderly and disabled people. *Proceedings of the 11th International Conference on Computers Helping People with Special Needs*, pages 114–122, 2008. 4

[26] Marilyn Cash. At home with at (assistive technology), research report, `http://www.dementia\-voice.org.uk/Projects/At_Home_with_AT_main.pdf`, 2004. 9

[27] Sung Hyuk Cha, Sungsoo Yoon, and C.C. Tappert. On binary similarity measures for handwritten character recognition. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, **1**, pages 4–8, 2005. 28, 88, 89

[28] Marie Chan, Daniel Estève, Christophe Escriba, and Eric Campo. A review of smart homes-present state and future challenges. *Comput. Methods Prog. Biomed.*, **91**:55–81, July 2008. 8

[29] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, **41**, 2009. 84, 86

[30] Liming Chen and Ismail Khalil. Activity recognition: Approaches, practices and trends. *in book: Activity Recognition in Pervasive Intelligent Environments*, **4**:1–32, May 2011. 13

[31] Seung-Seok Choi, Sung-Hyuk Cha, and Charles C. Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, **8**[1]:43–48, 2010. 89

[32] Pau-Choo Chung and Chin-De Liu. A daily behaviour enabled hidden markov model for human behaviour understanding. *Pattern Recognition*, **41**:1589–1597, May 2008. 23

[33] Daniel S. Clouse, C. Lee Giles, Bill G. Horne, and Garrison W. Cottrell. Time-delay neural networks: Representation and induction of finite-state machines. *IEEE Transaction on Neural Networks*, **8**, September 1997. 39

[34] D. Cook. Learning setting-generalized activity mdoels for smart spaces. *IEEE Intelligent Systems*, 2011. xvii, 131, 132

[35] D. Cook and S. Das. Smart environments: Technology, protocols and applications. 2004. 2

[36] Diane J. Cook. Making sense of sensor data. *IEEE Pervasive Computing*, [6]:105–108, 2007. 3, 10, 17

[37] Diane J. Cook, Juan C. Augusto, and Vikramaditya R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, pages 277–298, 2009. 2, 55

[38] Diane J. Cook, Michael Youngblood, and Edwin O. Heierman. Mavhome: An agent-based smart home. In *Proceedings. of the First IEEE International Conference on Pervasive Computing and Communications (PERCOM 03)*, pages 521–524, 2003. 1, 2

[39] P'adraig Cunningham. Dimension reduction. *Technical Report*, pages 1–24, August 2007. 97, 98

[40] Barnan Das, Chao Chen, Nairanjana Dasgupta, Diane Cook, and Adriyana M. Seelye. Automated prompting in a smart home environment. *IEEE International Conference on Data Mining Workshops*, pages 1045–1052, 2010. 24

[41] Sajal K. Das, Diane J. Cook, Amiya Bhattacharya, Edwin O. Heierman, and Tze Y. Lin. The role of prediction algorithms in the mavhome smart home architecture. *IEEE Wireless Communications*, **9**:77–84, 2002. 20

[42] Howard Demuth, Howard Demuth, and Martin Hagan. Neural network toolbox 6 user's guide. *The MathWorks*, 2008. 40

[43] Cook Diane and Tseng Chiu-Che. Mining from time series human movement data. *IEEE International Conference on Systems, Man and Cybernetics*, **13**:3241–3243, 2006. 17

[44] Anh Dinh, Daniel Teng, Li Chen, Yang Shi, Carl McCrosky, Jenny Basran, and Vanina Del Bello−Hass. Implementation of a physical activity monitoring system for the elderly people with built-in vital sign and fall detection. *Sixth International Conference on Information Technology: New Generations*, pages 1226–1231, April 2009. 57

[45] Faiyaz Doctor, Hani Hagras, and Victor Callaghan. A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *IEEE Transaction on Systems, Man, and Cybernetics-part A: Systems and Humans*, **35**:55–65, January 2005. 2, 19, 53, 54

[46] Georg Dorffiner. Neural networks for time series processing. *Neural Network World*, **4**[96]:447–468, 1996. 31, 32, 38

[47] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, **1**, pages 838–845, June 2005. 14, 15, 23

[48] Jeffery L. Elman. Finding structure in time. *Cognitive science*, **14**:179–116, 1990. 44

[49] Zippy Erlich, Roy Gelbard, and Israel Spiegler. Data mining by means of binary representation:a model for similarity and clustering. *Information Systems Frontiers*, **4**:187–197, 2002. 28

[50] Sumi Helal Eunju Kim and Diane Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, **9**:48–53, January 2010. 13, 14, 29

[51] Mark Hawley Fabien Cardinaux, Simon Brownsell and David Bradley. Modelling of behavioural patterns for abnormality detection

in the context of lifestyle reassurance. *Proc. CIARP 2008, Havana, Cuba*, pages 243–251, 2008. 11, 22, 65, 71, 73

[52] Michael Falk, Frank Marohn, Ren′e Michel, Daniel Hofmann, and Maria Macke. A first course on time series analysis: Examples with sas. *SAS Institute Inc.*, 2006. 31

[53] Holmes Finch. Comparison of distance measures in clusteranalysis with dichotomous data. *Journal of Data Science*, **3**:85–100, 2005. 88

[54] Martin Floeck, Lothar Litz, and Thorsten Rodner. An ambient approach to emergency detection based on location tracking. In *Proceedings of the 9th international conference on Toward useful services for elderly and people with disabilities: smart homes and health telematics*, ICOST'11, pages 296–302, Berlin, Heidelberg, 2011. Springer-Verlag. 15

[55] R. J. Frank, N. Davey, and S. P. Hunt. Time series prediction and neural networks. *J. Intell. Robot. Syst.*, **31**:91–103, 2001. 17

[56] Virone G. and Sixsmith A. Monitoring activity patterns and trends of older adults. *30th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, pages 2071–2074, 2008. 63

[57] A. I. Galarniotis, A. C. Tsakoumis, P. Fessas, S. S. Vladov, and V. M. Mladenov. Using elman and fir neural networks for short term electric load forecasting. In *Signals, Circuits and Systems, 2003. SCS 2003. International Symposium on*, **2**, pages 433–436, July 2003. 45

[58] Diane J. Cook Gaurav Jain and Vikramaditya Jakkula. Monitoring health by detecting drifts and outliers for a smart environment inhabitant. In *4th International Conference on Smart Homes and Health Telematics*, June 2006. 24, 66, 67

[59] Arpad Gellert and Lucian Vintan. Person movement prediction using hidden markov models. *Studies in Informatics and Control*, **15**:17–30, March 2006. 14, 35, 145

[60] Felix A. Gers, Douglas Eck, and Jurgen Schmidhuber. Applying lstm to time series predictable through time-window approaches. *Lecture notes in Computer Science*, pages 669–676, 2001. 49

[61] Wei Yee Goh, Chee Peng Lim, and Kok Khiang Peh. Predicting drug dissolution profiles with an ensemble of boosted neural networks: a time series approach. *Neural Networks, IEEE Transactions on*, **14**[2]:459–463, March 2003. 45

[62] Fabio A. González and Dipankar Dasgupta. Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines*, **4**:383–403, December 2003. 88

[63] Manabu GOUKO and Koji ITO. An action generation model using time series prediction. In *Proceedings of International Joint Conference on Neural Networks, USA*, pages 602–607, 2007. 32

[64] A. Gustavsson, L. Jonsson, R. McShane, M. Boada, A. Wimo, and AS. Zbrozek. Willingness-to-pay for reductions in care need: estimating the value of informal care in alzheimer's disease. *Int J Geriatr Psychiatry*, September 2009. 4, 9

[65] J. Boudy H. Medjahed, D. Istrate and B. Dorizzi. A fuzzy logic system for home elderly people monitoring (emutem). *10th WSEAS International Conference on Fuzzy Systems (FS'09)*, pages 69–75, 2009. 10

[66] J. Boudy H. Medjahed, D. Istrate and B. Dorizzi. A fuzzy logic system for home elderly people monitoring (emutem),. *10th WSEAS International Conference on Fuzzy Systems (FS'09)*, pages 69–75, 2009. 19, 102

[67] Tareq Hadidi and Norbert Noury. A predictive analysis of the night-day activities level of older patient in a health smart home. In *Ambient Assistive Health and Wellness Management in the Heart of the City*, **5597** of *Lecture Notes in Computer Science*, pages 290–293. Springer Berlin / Heidelberg, 2009. 16

[68] H. Hagras. Embedding computational intelligence in pervasive spaces. *IEEE Pervasive Computing*, **6**:85–89, 2007. 9

[69] Raffay Hamid, Siddhartha Maddi, Amos Johnson, Aaron Bobick, Irfan Essa, and Charles Isbell. A novel sequence representation for unsupervised analysis of human activities. *Elsevier Science, Artificial Intelligence*, **173**[14]:1221–1244, 2009. xiii, 11

[70] James Hammerton. Clause identification with long short term memory. In *Proceedings of CoNLL-2001*, pages 61–63, 2001. 48

[71] Jiawei Han and Micheline Kamber. Data mining Concepts and techniques. *Elsevier Inc. Publishing*, 2006. 97, 98

[72] Min Han, Jianhui Xi, Shiguo Xu, and Fu-Liang Yin. Prediction of chaotic time series based on the recurrent predictor neural network. *Signal Processing, IEEE Transactions on*, **52**[12]:3409–3416, December 2004. 18, 29, 44

[73] M.R. Hassan and B. Nath. Stock market forecasting using hidden markov model: a new approach. In *5th International Conference on Intelligent Systems Design and Applications*, pages 192–196, September 2005. 35

[74] Tamara Hayes, Misha Pavel, and Jeffrey Kaye. An approach for deriving continuous health assessment indicators from in-home sensor data. *Technology and aging, IOS Press*, pages 130–137, 2008. 66

[75] A. Helal, J. King, and R. Bose et al. Assistive environments for successful aging. *Advanced Intelligent Environments, Achilles D. Kameas, Victor Callagan, Hani Hagras, Michael Weber and Wolfgang Minker (Eds.) in , Springer US*, 2009. 9

[76] Sven Hellbach, Soren Strauss1, Julian P. Eggert, Edgar Korner, and Horst-Michael Gross. Echo state networks for online prediction of movement data - comparing investigations. **5163**:710–719, 2008. 48

[77] Yu Hen Hu and Jenq-Neng Hwang. Handbook of neural network signal processing. *CRC Press*, 2002. 33, 35, 37, 44, 51

[78] Anna Huang. Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand*, pages 49–56, 2008. 28

[79] Hsu Hui-Huang, Lu Kun-Chi, and M. Takizawa. Abnormal behaviour detection with fuzzy clustering for elderly care. *Computer Symposium*, pages 6–11, December 2010. 12, 25

[80] Sebastian Lauhr Hung and H. Bui Svetha Venkatesh Geoff A. W. West. Recognition of human activity through hierarchical stochastic learning. In *In International Conference on Pervasive Computing and Communication*, 2003. 3, 14

[81] Sajid Hussain, Senol Erdogen, and Jong Park. Monitoring user activities in smart home environments. *Information Systems Frontiers*, **11**:539–549, 2009. 20

[82] Rob J Hyndman. Moving averages, 2009. 77

[83] F. Rivera Illingworth, V. Callaghan, and H. Hagras. Detection of normal and novel behaviours in ubiquitous domestic environments. *The Computer Journal*, **53**:142–151, 2010. 4

[84] R. Illingworth, V. Callagha, and H. Hagras. A neural network agent based approach to activity detection in ami environments. In *IEE International Workshop, Intelligent Environments (IE05)*, pages 1–12, 2005. 18

[85] R. Illingworth, V. Callagha, and H. Hagras. Towards the detection of temporal behavioural patterns in intelligent environments. In *2nd IET International Conference on Intelligent Environments*, pages 119–125, 2006. 18, 25

[86] B. Ivanov, H. Ruser, and M. Kellner. Presence detection and person identification in smart homes. In *International Conference Sensors and Systems. State Technical University Saint-Petersburg Russia*, pages 80–85, 2002. 1

[87] Herbert Jaeger. A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the echo state network approach. *GMD Report 159, German National Research Centre for Information Technology*, **18**[1]:1–48, 2002. 45, 148

[88] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *SCIENCE*, **304**, April 2004. 32, 46, 48

[89] Vikramaditya Jakkula. Predictive data mining to learn health vitals of a resident in a smart home. *Seventh IEEE International Conference on Data Mining – Workshops*, pages 163–168, October 2007. 18

[90] Vikramaditya Jakkula and Diane Cook. Outliers detection in smart environment structured power datasets. In *Proceedings of the 2010 Sixth International Conference on Intelligent Environments*, IE '10, pages 29–33, 2010. 24, 28

[91] Vikramaditya Jakkula and Diane J. Cook. Detecting anomalous sensor events in smart home data for enhancing the living experience. *Proceedings of the AAAI Workshop on Artificial Intelligence and Smarter Living : The Conquest of Complexity*, 2011. 26, 83, 85

[92] Vikramaditya Jakkula, Diane J. Cook, and Aaron S. Crandall. Temporal pattern discovery for anomaly detection in a smart home. In *3rd IET International Conference on Intelligent Environments*, pages 339–345, 2007. 25

[93] Vikramaditya R. Jakkula, G. Michael Youngblood, and Diane J. Cook. Identification of lifestyle behaviour patterns with prediction of the happiness of an inhabitant in a smart home. In *AAAI Workshop on Computational Aesthetics: Artificial Intelligence Approaches to Beauty and Happiness, Boston*, July 2006. 20

[94] J. S. R. Jang, C. T. Sun, and E. Mizutani. Neurofuzzy and soft computing a computational approach to learning and machine intelligence. *Prentice Hall, Inc.*, 1997. 103

[95] Orlando De Jesus and Martin T. Hagan. Backpropagation algorithms for a broad class of dynamic networks. *IEEE Transaction on Neural networks*, **18**[1]:14–27, 2007. 46

[96] Chuanjin Jiang and Fugen Song. Forecasting chaotic time series of exchange rate based on non-linear autoregressive model. In *2nd International Conference on Advanced Computer Control (ICACC)*, **5**, pages 238–241, 2010. 41, 42

[97] I.T. Jolliffe. Principal component analysis. *Springer; 2nd ed. edition*, 2002. 77, 106

[98] P. Junior Jose Maria and Guilherme A. Barreto. Long-term time series prediction with the narx network: An empirical evaluation. *Neurocomputing*, **71**:3335–3343, March 2007. 41, 44

[99] JustChecking. Supporting independence people with dementia, `http://www.justchecking.co.uk`, January 2011. 9, 58, 59

[100] Rachid Kadouche, Hélène Pigot, Bessam Abdulrazak, and Sylvain Giroux. User's behaviour classification model for smart houses occupant prediction. *Atlantis Ambient and Pervasive Intelligence*, **4**:149–164, 2011. 19, 58

[101] Wonjoon Kang, Dongkyoo Shin, and Dongil Shin. Detecting and predicting of abnormal behavior using hierarchical markov model in smart home network. In *Industrial Engineering and Engineering Management (IE EM), 2010 IEEE 17Th International Conference on*, pages 410–414, October 2010. 22

[102] Henry Kautz, Oren Etzion, Dieter Fox, and Dan Weld. Foundations of assisted cognition systems, technical report. 2003. 15

[103] A. Kenner. Securing the elderly body: Dementia, surveillance, and the politics of aging in place. *Surveillance and Society*, **5**:252–269, 2008. 9

[104] Choong Hun Kim, Kyungseo Park, Jack Fu, and Ramez Elmasri. Architectures for streaming data processing in sensor networks. *AICCSA*, pages 1–59, 2005. 3

[105] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1–7, 1995. 115

[106] Sachin Kumar, Vasilis Sotiris, and Michael Pecht. Health assessment of electronic products using mahalanobis distance and projection pursuit analysis. *International Journal of Computer and Information Engineering*, **2**:243–251, 2008. 101

[107] Tze Leung Lai and Samuel PO-Shing Wong. Stochastic neural networks with applications to nonlinear time series. *Journal of the American Statistical Association*, **96**[455], 2001. 33

[108] Gad Landau, Abraham Bookstein, Shmuel Tomi Klein, and Timo Raita. Fuzzy hamming distance: A new dissimilarity measure (extended abstract). In *Combinatorial Pattern Matching*, **2089** of *Lecture Notes in Computer Science*, pages 86–97. Springer Berlin / Heidelberg, 2006. 90

[109] Hui Li, Qingfan Zhang, and Peiyong Duan. A novel one-pass neural network approach for activities recognition in intelligent environments. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 50–54, June 2008. 3, 14, 15, 17

[110] Hong Liangyou, Jiang Dongxiang, Huang Qian, and Ding Yongshan. Wind speed forecasting using fully recurrent neural network in wind power plants. *IEEE Transactions on Energy Conversion*, **11**:762–767, 2002. 32

[111] T.Warren Liao. Clustering of time series data a survey. *Pattern Recognition*, **38**:1857–1874, 2005. 85, 86

[112] Jessica Lin, Michail Vlachos, Eamonn Keogh, and Dimitrios Gunopulos. Iterative incremental clustering of time series. pages 106–122, 2004. 84, 85

[113] Tsung-Nan Lin, C. L. Giles, B. G. Horne, and Sun-Yuan Kung. A delay damage model selection algorithm for narx neural networks. *IEEE Transactions on Signal Processing*, **45**[11]:2719–2730, November 1997. 41

[114] Jun Liu, Khiang-Wee Lim, Rajagopalan Srinivasan, and Xuan-Tien Doan. On-line process monitoring and fault isolation using pca. *Proceedings of the IEEE International Symposium on Intelligent Control Limassol, Cyprus*, pages 27–29, June 2005. 97, 100, 101, 104

[115] J. Loane, B. O'Mullane, and B. Bortz andR.B. Knapp. Interpreting presence sensor data and looking for similarities between homes using cluster analysis. *5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, page 438, May. 25

[116] Fernando Loureno, Victor Lobo, and Fernando Bacao. Binary-based similarity measures for categorical data and their application in self-organizing maps. pages 1–18, 2004. 88

[117] Sebastian Luhr, Svetha Venkatesh, Geoff West, and Hung H. Bui. Duration abnormality detection in sequences of human activity. *Technical report, Department of Computing*, pages 117–132, May 2004. 23

[118] P. Manjunatha, A. K. Verma, and A. Srividya. Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method. *Proceeding of IEEE Region 10 Colloquium the 3rd International Conference on Industrial Information System (ICIIS)*, pages 8–10, December 2008. 19

[119] F. D. Marques, L. de F. Rodrigues de Souza, D. C. Rebolho, A. S. Caporali, and E. M. Belo. Application of time-delay neural and recurrent neural networks for the identification of a hingeless helicopter blade flapping and torsion motions. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, **XXVII**, April-June 2005. 38, 39

[120] Trevor Martin, Basim Majeed, Beum-Seuk Lee, and Nick Clarke. Fuzzy ambient intelligence for next generation tele-care. *IEEE International Conference on Fuzzy Systems; Canada*, pages 16–21, July 2006. 27, 101

[121] K. Matsuoka, H. Yamamoto, S. Inoue, S. Tanaka, K. Ohta, Y. Matsumura, T. Kometani, and K. Fukushima. Smart house monitoring living behaviours for finding unusual conditions. *In Proceedings of ROBOMEC04*, pages 1–4, 2004. 26

[122] P. J. Mccullagh, W. Carswell, J. C. Augusto, S. Martin, M. D. Mulvenna, H. Zheng, H. Y. Wang, J. G. Wallace, K. Mcsorley, B. Taylor, and W. P. Jeffers. State of the art on night-time care of people with dementia. *IET Assisted Living Conference, London, UK*, pages 24–25, March 2009. 9

[123] Hamid Medjahed, Dan Istrate, Jerome Boudy, and Bernadette Dorizzi. Human activities of daily living recognition using fuzzy logic for elderly home monitoring. *FUZZ IEEE Korea*, pages 20–24, August 2009. 19, 27, 102, 103

[124] Rachel E. Reilly Michael D. Lee and Marcus A. Butavicius. An empirical evaluation of chernoff faces, star glyphs, and spatial visualizations for binary data. *In CRPITS 24:Proceedings of the Australian symposium on Information visualisation*, pages 1–10, 2003. 19, 27, 75, 102

[125] Wolfgang Mller, Thomas Nocke, and Heidrun Schumann. Enhancing the visualization process with principal component analysis to support the exploration of trends. *Conferences in Research and Practice in Information Technology (CRPIT), Tokyo, Japan*, **60**, February 2006. 77, 97, 98

[126] Dorothy N. Monekosso and Paolo Remagnino. Anomalous behaviour detection: Supporting independent living. *Advanced Information and Knowledge Processing, Springer*, pages 33–48, 2009. 3, 10, 14, 22

[127] Taketoshi Mori, Akinori Fujii, Masamichi Shimosaka, Hiroshi Noguchi, and Tomomasa Sato. Typical behaviour patterns extraction

and anomaly detection algorithm based on accumulated home sensor data. *Future Generation Communication and Networking*, **2**:12–18, 2007. 22, 52

[128] M. Mozer. The neural network house: An environment that adapts to its inhabitants. In *Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, pages 110–114, 1998. 17

[129] Richi Nayak and Paul te Braak. Temporal pattern matching for the prediction of stock prices. *Proceedings of the 2nd international workshop on Integrating artificial intelligence and data mining (AIDM)*, **84**, December 2007. 31

[130] Nist/Seatech. e-handbook of statistical methods, `http://www.itl.nist.gov/div898/handbook/eda/section3/autocopl.htm`. 66

[131] D. Novák, K. Mucha, and T. Al-Ani. Long short-term memory for apnea detection based on heart rate variability. *30th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, [20-25]:5234–5237, 2008. 48

[132] Oliver Obst, X. Rosalind Wang, and Mikhail Prokopenko. Using echo state networks for anomaly detection in underground coal mines. *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 219–229, 2008. 48

[133] M. Hamel P. Boissy, S. Choquette and N. Noury. User-based motion sensing and fuzzy logic for automated fall detection in older adults. *Telemedicine Journal and e-Health*, **13**:683–693, 2007. 9

[134] Ajoy K. Palit and Dobrivoje Popovic. Computational intelligence in time series forecasting: Theory and engineering applications. *Springer*, 2005. 32

[135] Zhijian Pan. Principal component analysis based visualization and human melanoma classification. *Dimension Contemporary German Arts And Letters*, **2**:1–3, 2001. 77, 97, 98

[136] George Papamatthaiakis, George C. Polyzos, and George Xylomenos. Monitoring and modelling simple everyday activities of the elderly at home. In *Proceedings of the 7th IEEE conference on Consumer communications and networking conference*, CCNC'10, pages 617–621. IEEE Press, 2010. 18, 54

[137] Kyungseo Park, Yong Lin, Vangelis Metsis, Zhengyi Le, and Fillia Makedon. Abnormal human behavioural pattern detection in assisted living environments. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '10, pages 91–98. ACM, 2010. 28

[138] Sang-Ho Park, Ju-Hong Lee, Jae-Won Song, and Tae-Su Park. Forecasting change directions for financial time series using hidden markov model. *Lecture Notes in Computer Science, Springer*, **5589**:184–191, 2009. 14

[139] Wen-Chih Peng, WangTing Huang, YiLing Chen, and PeiChing Liao. Tagfree: Identifying users without tags in smart home environments. *IEEE*, 2007. 15

[140] J.C. Principe, J.-M. Kuo, and B. de Vries. Backpropagation through time with fixed memory size requirements. *Proceedings of the IEEE-SP Workshop Neural Networks for Signal Processing III.*, pages 207–215, September 1993. 45

[141] Jos C. Principe, Neil R. Euliano, and W. Curt Lefebvre. Neural and adaptive systems: Fundamentals through simulations. 2000. 40

[142] Protégé. The protégé ontology editor and knowledge acquisition system, `http://protege.stanford.edu/`, October. 144

[143] T. Adlam R. Faulkner R. Orpwood, C. Gibbs and D. Meeeahawatte. 9

[144] Sira Panduranga Rao and Diane J. Cook. Predicting inhabitant action using action and task models with application to smart homes. *In-*

ternational Journal on Artificial Intelligence Tools, **13**:81–100, 2004. 3, 14

[145] PARISA RASHIDI, DIANE J. COOK, LAWRENCE B. HOLDER, AND MAUREEN SCHMITTER-EDGECOMBE. Discovering activities to recognize and track in a smart environment. *IEEE Transactions on Knowledge and Data Engineering*, **23**:527–539, April 2011. 28

[146] SANJAY RAWAT, V. P. GULATI, ARUN K. PUJARI, AND V. RAO VEMURI. Intrusion detection using text processing techniques with a binary-weighted cosine metric. *Journal of Information Assurance and Security*, **1**:43–50, 2006. 28

[147] PAYAM REFAEILADEH, LEI TANG, AND HUAN LIU. Cross−validation. *Encyclopaedia of Database Systems (EDBS). Springer*, pages 1–6, November 2009. 114, 115

[148] LIOR ROKACH AND ODED MAIMON. Data mining and knowledge discovery handbook. *Springer-Verlag*, 2005. 89

[149] D. ROSENBERG. Trend analysis and interpretation. *Maternal and Child Health Information Resource Centre*, 1997. 63

[150] K. SAWAI AND M. YOSHIDA. Algorithm to detect abnormal states of elderly persons for home monitoring. *Systems and Computing in Japan.*, **38**:34–42, June 2007. 17

[151] JURGEN SCHMIDHUBER AND SEPP HOCHREITER. Long short term memory. *Neural Computation*, **9**[8]:1735–1780, 1997. 48, 49, 150

[152] SILVIA SELINSKI AND KATJA ICKSTADT. Similarity measures for clustering snp data. In *Technical report of university of Dortmund*, pages 1–63, 2005. 89

[153] A. SERNA, H. PIGOT, AND V. RIALLE. Modeling the progression of alzheimer's disease for cognitive assistance in smart homes. *User Modeling and User-Adapted Interaction*, **17**:415–438, 2007. 10

[154] Nigel Shadbolt. Ambient intelligence. *IEEE Intelligent Systems*, **18**:2–3, 2003. 54

[155] Zhiwei Shi and Min Han. Support vector echo-state machine for chaotic time-series prediction. *IEEE Transactions on Neural Networks*, **18**:359–372, March 2007. 47, 48

[156] Jae Hyuk Shin, Boreom Lee, and Kwang Suk Park. Detection of abnormal living patterns for elderly living alone using support vector data description. *IEEE Transaction on Information Technology in Biomedicine*, **15**:438–448, May 2011. 26

[157] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chan. A novel anomaly detection scheme based on principal component analysis. *in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03*, pages 172–179, 2003. 24

[158] Michael Siek and Dimitri Solomatine. Chaotic model with data assimilation using narx network. In *Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA*, pages 2516–2523, 2009. 41, 44

[159] Kelvin Sim, Clifton Phua, Ghim-Eng Yap, Jit Biswas, and Mounir Mokhtari. Activity recognition using correlated pattern mining for people with dementia. *33rd Annual International Conference of the IEEE EMBS Boston Massachusetts USA*, 2011. 16

[160] Geetika Singla, Diane J. Cook, and Maureen Schmitter-Edgecombe. Recognizing independent and joint activities among multiple residents in smart environments. *Journal of Ambient Intelligent Human Computing*, **1**:57–63, 2010. 3, 15

[161] Mark D. Skowronski and John G. Harris. 47

[162] Stefan Squartini, Arnir Hussain, and Francesco Piazza. A recurrent multi-scale architecture for long-term memory prediction task. *Speech, and Signal Processing*, **2**[6-10]:789–792, 2003. 18, 44

[163] George Stefatos and A. Ben Hamza. Cluster pca for outliers detection in high-dimensional data. *IEEE International Conference on Systems, Man and Cybernetics*, pages 3961–3966, October 2007. 24, 27

[164] J.P.F. Sum, W.-K. Kan, and G.H. Young. A note on the equivalence of narx and rnn. *Neural Computing and Applications*, **8**[1]:33–39, 1999. 41

[165] Huy Anh Nguyen Tam Van Nguyen and Deokjai Choi. Development of a context aware virtual smart home simulator. *Proc. of ICUT'08, Ho Chi Minh City, Vietnam*, 2008. 62

[166] Dante I. Tapia, Ajith Abraham, Juan M. Corchado, and Ricardo S. Alonso. Agents and ambient intelligence: case studies. *Journal of Ambient Intelligence and Humanized Computing*, **1**[2]:85–93, 2010. 10

[167] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. *PERVASIVE, LNCS, Springer, Heidelberg*, **3001**:158–175, 2004. 3, 14

[168] Steve Kell Tracy Barger, Majd Alwan and Beverely Turner. Objective remote assessment of activities of daily living: Analysis of meal preparation patterns. *Medical Automation Research Center, University of Virginia Health System*, 2002. 25

[169] An C. Tran, Stephen Marsland, Jens Dietrich, Hans W. Guesgen, and Paul Lyons. Use cases for abnormal behaviour detection in smart homes. In *Proceedings of the Aging friendly technology for health and independence, and 8th international conference on Smart homes and health telematics*, ICOST'10, pages 144–151. Springer-Verlag, 2010. 27

[170] Claudia Ulbricht. Multi recurrent network for traffic forecasting. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 883–888, 1994. 32

[171] T. van Kasteren, G. Englebienne, and B. Krse. An activity monitoring system for elderly care using generative and discriminative models. *Personal and Ubiquitous Computing*, **14**:489–498, 2010. 57

[172] M. Ventresca and H.R. Tizhoosh. Opposite transfer functions and backpropagation through time foundations of computational intelligence. *FOCI 2007. IEEE Symposium on*, pages 570–577, April 2007. 45

[173] Lucian Vintan, Arpad Gellert, Jan Petzold, and Theo Ungerer. Person movement prediction using neural networks. In *In First Workshop on Modeling and Retrieval of Context*, 2004. 17

[174] S. Vitabile, M. Pernice, and S. Gaglio. Daily peak temperature forecasting with elman neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, **4**, pages 2765–2769, July 2004. 45

[175] Thomas Voegtlin. Recursive self-organizing maps. *Neural Networks, Special Issue, Elsevier Science*, **15**:979–991, 2002. 49

[176] Thomas Voegtlin and Peter F. Dominey. Recursive self-organizing maps. *Advances in Self-Organizing Maps, Springer*, pages 210–215. 49, 51

[177] K. Vrotsou, K. Ellegard, and M. Cooper. Everyday life discoveries: Mining and visualizing activity patterns in social science diary data. In *Information Visualization, 11th International Conference*, pages 130–138, 2007. xiii, 10

[178] Eric A. Wan. Time series prediction by using a confectionist network with internal delay lines. *Time Series Prediction:Forecasting the Future and Understanding the Past, Addison-Wesley*, 1993. 32

[179] P. J. Werbos. Backpropagation through time what it does and how to do it. *Proceedings of the IEEE*, **78**[10]:1550–1560, 1990. 46

[180] Daniel Wilson and Chris Atkeson. Automatic health monitoring using anonymous binary sensors. *In Extended Proceedings of CHI 2004: Workshops Vienna, Austria*, pages 1719–1720, 2004. 15

[181] Daniel Wilson and Chris Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. pages 62–79, 2005. 2

[182] A. Wimo, B. Winblad, and L. Jonsson. An estimate of the total worldwide societal costs of dementia in 2005. *Alzheimers Dement.*, **3**:81–91, April 2007. 9

[183] Y. Wu, A. Ganapathiraju, and J. Picone. Baum-welch re-estimation of hidden markov model. *Institute for Signal and Information Processing*, pages 1–10, June 1999. 34

[184] Hang Xie, Hao Tang, and Yu-He Liao. Time series prediction based on narx neural networks: An advanced approach. In *Machine Learning and Cybernetics, 2009 International Conference on*, **3**, pages 1275–1279, July 2009. 42

[185] Hung2 Ya-Xuan, Chiang Chih-Yen, Steen J. Hsu, and Chia-Tai Chan. Abnormality detection for improving elder's daily life independent. In *Proceedings of the Aging friendly technology for health and independence, and 8th international conference on Smart homes and health telematics*, ICOST'10, pages 186–194. Springer-Verlag, 2010. 23

[186] Qiuyan Yan, Shixiong Xia, and Yilong Shi. An anomaly detection approach based on symbolic similarity. In *Control and Decision Conference*, May 2010. 11, 28

[187] J. Ye and S. Ji. Discriminant analysis for dimensionality reduction An overview of recent developments in biometrics Theory methods and applications eds n. v. boulgouris and k. n. plataniotis and e. michelitzanakou. *John Wiley & Sons Inc˙ Hoboken NJ USA*, 2009. 97

[188] Li Ye, Zhi-Guang Qin, Juan Wang, and Jing Jin. Anomaly event detection in temporal sensor network data of intelligent environments. In *Computer Engineering and Technology (ICCET), 2nd International Conference on*, **7**, pages 414–420, April 2010. 21, 28

[189] Jie Yin, Qiang Yang, and Jeffrey Junfeng Pan. Sensor-based abnormal human-activity detection. *IEEE Transaction on Knowledge and Data Engineering*, **20**:1082–1090, August 2008. 26

[190] Yong Yin and Kazuhiko Yasudab. Similarity coefficient methods applied to the cell formation problem: A taxonomy and review. *Int. J. Production Economics*, **101**:329–352, 2006. 113

[191] Osman Yogurtcu, Engin Erzin, and Attila Gursoy. Extracting gene regulation information from micro array time-series data using hidden markov models. *Lecture Notes in Computer Science, Springer*, **4263**:144–153, 2006. 14

[192] X. Yu. Approaches and principles of fall detection for elderly and patient. *10th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 42–47, 2008. 9

[193] GP Zhang and VL Berardi. Time series forecasting with neural network ensembles: an application for exchange rate prediction. *Journal of the Operational Research Society*, **52**:652–664, 2001. 17, 36, 38

[194] Huiru Zheng, Member, Haiying Wang, and Norman Black. Human activity detection in smart home environment with self-adaptive neural networks. In *IEEE International Conference on Networking, Sensing and Control*, pages 1505–1510, 2008. 17

Index