

Northumbria Research Link

Citation: Wonders, Martin (2016) Activity Recognition in Monitored Environments using utility Meter disaggregation. Doctoral thesis, Northumbria University.

This version was downloaded from Northumbria Research Link:
<http://nrl.northumbria.ac.uk/id/eprint/31614/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

Activity Recognition in Monitored Environments using utility Meter disaggregation

M A Wonders

PhD

2016

Activity Recognition in Monitored Environments using utility Meter disaggregation

Martin Anthony Wonders

A thesis submitted in partial fulfilment
of the requirements of the
University of Northumbria at Newcastle
for the degree of
Doctor of Philosophy
research undertaken in the
Faculty of Engineering and Environment

October 2016

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by the Faculty Ethics Committee, January 2012.

I declare that the Word Count of this Thesis is 45,556 words excluding references, table of contents, table of figures and title pages.

Name: Martin Wonders

Signature:

Date:

Abstract

Activity recognition in monitored environments where the occupants are elderly or disabled is currently a popular research topic and is being proposed as a possible solution that may help maintain the independence of an aging population within their homes, where these homes are adapted as monitored environments. Current activity recognition systems implement ubiquitous sensing or video surveillance techniques which inherently, to varying degrees, impinge on the privacy of the occupants of these environments. The research presented in this thesis investigates the use of Ubiquitous sensors within a smart home setting with a view to establishing whether activity recognition is possible with a reduced, less intrusive subset of sensors that can be realised using utility meter disaggregation techniques. The thesis considers the selection of sensors as a feature selection problem and concludes that data produced from water, electricity and PIR sensors contribute significantly to the recognition of selected activities. With an established method of activity recognition that implements a reduced number of sensors it can be argued that occupants of the monitored environment maintain a greater level of privacy. This level of privacy, however, is dependent on such systems being practically implementable into homes that are designed to assist and monitor the residents, and as such configuration and maintenance of these systems are also considered here. The utility meter disaggregation technique presented proves to perform exceptionally well when trained with large quantities of data, but gathering and labelling this data is, in itself, an intrusive process that requires significant effort and could compromise the practicality of such promising systems. This thesis considers methods for implementing synthesised, labelled training data for both disaggregation and activity recognition systems and shows that such techniques can significantly reduce the quantity of labelled training data required. The work presented shows a significant contribution, in the areas of sensor selection and the use of utility meter disaggregation for activity recognition, and also the use of synthesised labelled training data to reduce significant system training times. The work is carried out using a combination of publicly available datasets and data collected from a purpose built smart home which includes water and electricity meter disaggregation. It is shown that a system

for non-intrusive monitoring within an ambient environment, occupied by a single resident, is achievable using repurposed versions of the standard domestic infrastructure. More specifically it is demonstrated that a minimum baseline accuracy of 93.45% and F1-measure of 91.22 can be achieved using disaggregation at the water and electricity meters combined with locality context provided by home security PIR sensors. Methods of speeding up the deployment and commissioning process are proven to be viable, further demonstrating the potential practical application of the proposed system.

Acknowledgements

I would like to express my sincere appreciation to Prof. Fary Ghassemlooy for his patience and guidance through the early stages of this research. Without his encouragement at the beginning this work would never have been started. I would also like to thank Prof. Ahmed Bouridane for his guidance during the final year of the research, without which, this work would never have seen the light of day.

I also owe a debt of gratitude to Prof Alamgir Hossain for the encouragement and conviction of striving for the highest standards in the publication of research.

Finally I would like to thank Dr Graham Sexton for over 15 years of constant, professional, guidance and support that have culminated in the production of this thesis.

This work is dedicated to Fred and Wendy Wonders in the hope that it may make them proud.

Publications

“Training with synthesised data for disaggregated event classification at the water meter”, Expert Systems with Applications, Volume 43, pp. 15-22.

“Sensor selection for Human Activity Recognition in ambient environments using a Hybrid feature selection technique.”, to be submitted to IEEE Intelligent Systems.

“Human Activity Recognition using disaggregation at the utility meters.” to be submitted to Expert Systems with Applications.

“Synthetic Minority Oversampling using Gaussian Mixture Models to balance training data in Human activity recognition.”, to be published to IEEE Computational Intelligence.

Table of Contents

Chapter 1	Introduction	12
1.1	The Research Aims	14
1.2	Challenges in the field of Activity Recognition	16
1.2.1	Insufficient labelled data.....	16
1.2.2	Intra and inter-class variations.....	17
1.2.3	Multiple Occupancy	18
1.2.4	Privacy.....	18
1.3	Thesis structure and contributions	19
Chapter 2	Background	23
2.1	Definitions used in HAR.....	23
2.2	Classes of Activity.....	25
2.3	Approaches to HAR.....	26
2.3.1	Recognition of physical activity.....	27
2.3.2	Activity recognition using environmental sensors.....	30
2.3.3	Sensors and Activity definitions	40
2.4	Human Activity Recognition Summary	43
Chapter 3	Machine Learning for HAR	44
3.1	Machine Learning.....	45
3.1.1	Supervised Learning.....	47
3.2	Machine learning analysis framework.....	58
3.2.1	Data collection	59
3.2.2	Data pre-processing and Segmentation.....	62
3.3	Machine Learning Evaluation and Performance Metrics.....	66
3.4	Summary of machine learning in HAR	69
Chapter 4	Feature Selection for HAR.....	70
4.1	Feature Selection	71
4.1.1	Filter Methods.....	73
4.1.2	Wrapper Methods.....	74
4.1.3	Embedded Methods.....	74
4.2	Algorithm Evaluation	79
4.2.1	Algorithm Evaluation (Balanced dataset)	80
4.2.2	Algorithm Evaluation (Unbalanced dataset).....	82
4.3	Sensor analysis using feature selection	83

4.3.1	Feature selection analysis (Balanced dataset).....	85
4.3.2	Feature selection analysis (Unbalanced dataset)	86
4.4	Summary and Conclusion of Feature Selection for Activity Recognition	90
Chapter 5	Repurposed Smart Meters.....	92
5.1	Using a water meter as sensors.....	93
5.1.1	Related Work	93
5.1.2	Synthesising labelled training data	96
5.1.3	Comparison of ANN, SVM and KNN algorithms.....	97
5.1.4	Artificial Neural Networks (ANN).....	97
5.1.5	Support Vector Machine (SVM)	98
5.1.6	K-Nearest Neighbours (KNN)	100
5.1.7	The problem with labelled training data	100
5.2	Evaluation of Machine Learning Algorithms for water meter Disaggregation	101
5.3	Learning with synthesised training data	108
5.4	Experimental Results and Discussion.....	111
5.5	Summary and Conclusion of Repurposed smart meters	115
Chapter 6	HAR using sensor Fusion	117
6.1	Activity recognition in an Ambient Environment.....	118
6.2	The Reduced Sensor set.....	123
6.2.1	Data Collection.....	124
6.2.2	Data pre-processing and feature selection.....	126
6.3	The chosen activity set.....	126
6.3.1	Activity segmentation	128
6.4	Evaluation of Machine Learning algorithms	129
6.4.1	The addition of temporal features.....	133
6.4.2	Using disaggregated water data	136
6.5	Balancing training data with synthesis	138
6.5.1	Synthetic Minority Over-sampling (SMOTE)	138
6.5.2	Model evaluation using data from SMOTE	139
6.6	Summary of Activity Recognition using sensor Fusion	144
Chapter 7	Conclusions and future work	146
7.1	Future work.....	149
References		150

List of Figures

Figure 3.1: <i>Off-line Machine Learning Framework.</i>	58
Figure 3.2 Daily activities generated from the Tapia Data set. Data Source (Tapia, et al., 2004).	64
Figure 3.3 Daily activities generated from the CASAS Dataset. Data Source (Cook & Schmitter-Edgecombe, 2009)	64
Figure 4.1: <i>Activity counts for balanced dataset. Data source: (Crandall & Cook, 2008).</i>	80
Figure 4.2: <i>Activity counts before and after reduction. Data source (Tapia, et al., 2004)</i>	82
Figure 4.3 <i>Logistic Regression with L1 penalty. Data Source (Cook & Schmitter-Edgecombe, 2009)</i>	84
Figure 4.4 <i>Mean Sensor Importance. Data Source (Cook & Schmitter-Edgecombe, 2009)</i>	85
Figure 4.5 <i>Sensor Importance SVM with linear kernel. Data Source (Tapia, et al., 2004)</i>	87
Figure 4.6 <i>Sensor subset Performance SVM with linear kernel. Data Source (Tapia, et al., 2004)</i>	88
Figure 4.7 <i>Sensor subset Importance. Data Source (Tapia, et al., 2004)</i>	88
Figure 4.8 <i>Sensor subset counts - SVM with linear kernel. Data Source (Tapia, et al., 2004)</i>	89
Figure 4.9 <i>Electrical sensors from the 38 sensor subset. Data Source (Tapia, et al., 2004)</i>	89
Figure 5.1 <i>Flow event signature representing combined toilet flush and wash basin usage.</i>	101
Figure 5.2 <i>Precision chart for SVM, ANN and KNN classifiers</i>	106
Figure 5.3 <i>Recall chart for SVM, ANN and KNN classifiers</i>	107
Figure 5.4 <i>F1 chart for SVM, ANN and KNN classifiers</i>	107
Figure 5.5 <i>Accuracy chart for SVM, ANN and KNN classifiers</i>	108
Figure 5.6 <i>Precision chart using synthesised training data.</i>	111
Figure 5.7 <i>Recall chart using synthesised training data.</i>	112
Figure 5.8 <i>F1 chart using synthesised training data.</i>	112
Figure 5.9 <i>Accuracy chart for synthesised training data.</i>	113
Figure 5.10 <i>Screen shot from online water disaggregation meter</i>	114
Figure 6.1: <i>Ambient environment floor plans showing only those sensors used in the study.</i>	120
Figure 6.2: PIR and ground truth, water meter sensor counts for 7 week monitoring period.....	123
Figure 6.3: Electrical sensor counts for 7 week monitoring period	123
Figure 6.4: <i>Sample of an XML file showing a single Microwave event.</i>	124
Figure 6.5: <i>Comma-separated value files showing Light, Activity and PIR data.</i>	125
Figure 6.6: <i>Sample of an XML file showing a single en-suite wash basin event.</i>	125
Figure 6.7 <i>Activities chosen and counts over the monitoring period</i>	127
Figure 6.8 <i>Gantt chart of ground truth activity labels over the 7 week monitoring period.</i>	128
Figure 6.9: <i>Training and Test set split of activities.</i>	130
Figure 6.10: <i>Test set confusion matrix for the SVM classifier using a linear kernel.</i>	131
Figure 6.11: <i>Confusion matrix for Logistic Regression classifier using an L2 penalty.</i>	131

Figure 6.12: <i>Confusion matrix for the Random Forest classifier using gini as the split criterion.</i>	132
Figure 6.13: <i>Confusion matrix for the AdaBoost, DT classifier using entropy as the split criterion.</i>	132
Figure 6.14: <i>Activity counts at each hour of the day for the entire dataset.</i>	133
Figure 6.15: <i>Confusion matrix for the SVM classifier using a linear kernel and an "Hour of Day" feature.</i>	134
Figure 6.16: <i>Confusion matrix for Logistic Regression classifier using an L2 penalty and an "Hour of Day" feature.</i>	135
Figure 6.17: <i>KNN water meter sensor counts for 7 week monitoring period.</i>	136
Figure 6.18: <i>Confusion matrix for Logistic Regression classifier using an L1 penalty and using water disaggregation with a KNN classifier.</i>	137
Figure 6.19: <i>Parallel coordinate plot showing sensor counts for each activity in the training set.</i>	139
Figure 6.20: <i>Parallel coordinate plot showing sensor counts for each activity after SMOTE data synthesis of training data.</i>	140
Figure 6.21: <i>Confusion matrix for Logistic Regression classifier using an L2 penalty and water disaggregation and SMOTE.</i>	141
Figure 6.22: <i>Confusion matrix for a linear SVM classifier using water disaggregation and SMOTE.</i>	142
Figure 6.23: <i>Gantt chart showing 18 days of ground truth activity data used for algorithm testing.</i>	143
Figure 6.24: <i>Gantt chart showing 18 days of activity data as predicted by the LR classifier using an L2 penalty, trained on data that was balanced using SMOTE, where water utility data was produced using KNN disaggregation.</i>	144

List of Tables

Table 2.1: <i>List of activities and group classifications</i> (Atallah, et al., 2011)	28
Table 3.1 <i>Sample of data from</i> (Tapia, et al., 2004).....	60
Table 3.2 <i>Sample of data from</i> (Cook & Schmitter-Edgecombe, 2009).....	61
Table 3.3 <i>Sensor count representation of sample data from</i> (Tapia, et al., 2004)	65
Table 3.4 <i>Sensor count representation of sample data from</i> (Cook & Schmitter-Edgecombe, 2009)	66
Table 4.1 <i>Machine Learning classifier performance: Data source</i> (Crandall & Cook, 2008).....	81
Table 4.2 <i>Machine Learning classifier performance. Data source</i> (Tapia, et al., 2004).....	83
Table 4.3 <i>Classifier performance using sensor subset. Data source</i> (Crandall & Cook, 2008)....	86
Table 5.1 <i>Event name and quantity of clean events captured</i>	102
Table 5.2 <i>Sample values that make up the majority of the 376 features of a shower event</i>	103
Table 5.3 <i>SVM kernel accuracy using 2-fold cross validation and 100 features</i>	104
Table 5.4 <i>ANN accuracy using 2-fold cross validation and 100 features</i>	105
Table 5.5 <i>KNN Accuracy using 2-fold cross validation and 100 features</i>	105
Table 5.6 <i>Summary of best performing model trained and tested with real data</i>	105
Table 5.7 <i>Event type with the quantity of original real events, and the quantity of synthesised events synthesised from only 2 of the extreme data points from originals</i>	110
Table 5.8 <i>Summary of the best performing model trained with synthesised data and tested with real data</i>	111
Table 6.1: <i>Sample of labelled activity data using sensor count as features</i>	126
Table 6.2: <i>Machine learning model evaluation using activity data collected over 7 weeks</i>	130
Table 6.3: <i>Machine learning model evaluation with added “Hour of Day” feature</i>	134
Table 6.4: <i>Machine learning model evaluation using activity data with disaggregated water data using a KNN classifier</i>	137
Table 6.5: <i>Machine learning model evaluation using training data synthesised using the SMOTE technique</i>	141

List of Acronyms

Activities of Daily Living (IADL), 25
Ambient Intelligence (Aml), 14
Artificial Neural Network (ANN), 21
Centre for Advanced Studies in Adaptive Systems (CASAS), 32
Decision Trees (DT), 49
Dynamic Bayesian Network (DBN), 73
False Negatives (FN), 67
Feature Space Remapping (FSR), 39
Finite State Machines (FSM), 42
Hidden Markov models (HMMs), 32
Human Activity Recognition (HAR), 13
Intel Next Unit of Computing (NUC), 119
K-Nearest Neighbours (KNN), 100
Logistic Regression (LR), 49
Managing An intelligent Versatile (MAV), 34
Markov Chains (MCs), 32
Message Queuing Telemetry Transport (MQTT), 119
Mobile and Pervasive Computing Laboratory (MPCL), 31
Multi Home Transfer Learning (MHTL), 38
Naïve Bayes (NB), 35
National Health Service (NHS), 12
Radial basis function (RBF), 99
Radio Frequency Identification (RFID), 31
Random Forests (RF), 49
Regularised Logistic Regression (RLR), 76
Support Vector Machine (SVM), 21
Synthetic Minority Over-sampling (SMOTE), 118
television (TV), 35
True Negatives (TN), 66
True Positives (TP), 66
voice over IP (VOIP), 32
water closet (WC), 101

Chapter 1 Introduction

According to a report commissioned by Ofcom in 2010 (Lewin, et al., 2010), there were around 1.8 million people with moderate to severe disabilities in the UK, with 72% of these aged 65 or above. The same report predicts that the requirements for health care within the home will increase by 16% over the next 20 years. The 2015 key issues for parliament report (Young, 2015) states that between 2015 and 2020 the general population of the UK is expected to rise by 3%, the population of people over 65 is expected to increase by 12%, the population of people over 85 by 18% and the population of people aged 100 or above by 40%. Currently from an estimated UK population of around 61 million, 11.5 million people are of pensionable age and 4 million are living alone. As the UK population ages these numbers are expected to increase and will undoubtedly put a substantial strain on care providers, whether formal i.e. National Health Service (NHS) or informal i.e. family, friends or neighbours. According to a 2009 Help the Aged report (Help The Aged, 2009) an increase in divorce rates, as well as smaller family units and the increased global mobility in family members, will make it much more difficult to maintain the current input from those involved in informal health care.

Technology in the domestic environment is proposed as one of the strategies that can help to minimise the impact, as well as maintain the independence of an ageing population, and by incorporating more and more computational technology to produce smarter living environments, there exists the potential to provide more flexible assisted living spaces for people with the special needs that come with age or disability. Many of the studies carried out in this area, for example (Kumar & John, 2016), (Benmansour, et al., 2016), (Cook & Krishnan, 2015), (Steinhauer, et al., 2010) and (Logan, et al., 2007) focus around the use of ubiquitous sensitised spaces or video based surveillance and use activity recognition systems to interpret the data. These methods are generally intrusive and require expensive computational input as well as a high level of deployment impracticality.

The aim of this research is to investigate a system for non-intrusive monitoring of the elderly, or those with special needs, that can potentially form a practical implementation of use within assisted living environments that can aid in Human Activity Recognition (HAR) and so monitor

the behaviour of a resident. The proposed system incorporates a method of sensor fusion that implements disaggregation techniques, at both the water (Froehlich, et al., 2009) and electricity meters (Gupta, et al., 2010), to infer the activities of the inhabitants. The data from each meter is complemented with contextual, temporal data from movement sensors positioned in each room. Such a system can be implemented by utilising more computationally advanced versions of existing components of the home infrastructure, for example an unobtrusive monitoring system can be created by repurposing, the meter points and standard alarm system sensors with minimal disruption to the occupant. The advantages of the proposed system include reduced implementation time and effort, as the existing utility meter upgrade is the only requirement in terms of sensors used. Reduced training times as synthesis of training data reduces the quantity of labelled data required from the environment, this in turn leads to less invasive systems.

HAR has emerged as an active area of research over the past decade and finds context in many applications and fields such as video surveillance (Lin, et al., 2008), Human Computer Interaction (Zhang, 2012), health care (Zhu & Sheng, 2011) as well as personal proactive health monitoring (Zhang & Sawchuk, 2013). The purpose of any activity recognition system is to recognize the actions of an agent from a series of sensor observations using computational intelligence. For example in the work produced by (Lin, et al., 2008) the actions are categorised as walking, running, fighting, active and inactive, the agent in this system is a Human and the sensor observations are a sequence of frames taken from short video clips. By contrast in the research produced by (Zhang & Sawchuk, 2013) the agent is again a human and the activities are walk forward, walk left, go upstairs, jump etc. but the sensor observations are taken from a three axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer. Specifically these systems of activity recognition are referred to as HAR systems and are a multidisciplinary branch of activity recognition research that leverage the fields of sensor systems, ubiquitous computing, data mining, machine learning and data analysis with the aim of inferring the specific activity of the human in question. Once inferred, this activity can be used in the specific context of an application for example in (Lin, et al., 2008) the application context may be to determine whether there is a breach of security on a premises. The application context in (Zhang, 2012) could be to

interact with a computer software application or game. The context in (Zhu & Sheng, 2011) could be to determine if an elderly person is in need of assistance or in (Zhang & Sawchuk, 2013) it could be to inform the user of their calorific output throughout the day.

It follows that HAR system can be viewed as just one part of a more general ambient environment (Emiliani & Stephanidis, 2005) dedicated to the field of assisted living. Ambient environments take advantage of the ever increasing use of small scale, low cost sensing, computing, and network equipment to create computational intelligence within the environments that work with the occupant to enhance their daily lives. A fundamental aspect of ambient environments, based on the description and definition provided by (Juan Carlos Augusto, 2007) where the term Ambient Intelligence (AmI) is used, is that they are designed and built to interact with the inhabitant or user of the environment and thus there is an inherent requirement to establish and monitor the activities of that user.

An example high level design of an AmI system is described by (Augusto, et al., 2010) and places the HAR system (pose/activity and detection) within a processing layer between a sensor network and a middleware tier that indicates a level of knowledge accumulation, validation and user input.

1.1 The Research Aims

The previous section provided a brief, high level overview that placed the role of HAR systems within the more general field of ambient intelligent systems, and the high-level reasoning and visualisation components shown in (Augusto, et al., 2010) hint at a dependency or relationship to the use-case of a particular system. For example the activities to be recognised for a security system may be fundamentally different from the activity requirements for an ambient assisted living environment and so the context of the use case is of fundamental importance when developing and evaluating such systems.

One method of establishing a use-case context is to consider a combination of what is termed the “Context Quintet” (Brooks, 2003) of “when”, “where”, “what”, “who” and “why”:

- **When:** Temporal awareness, a system maintains some representation of time and seeks to record when events and activities occur, e.g. each event within the system may be represented by a time stamp of the interactions between the agents and the system.
- **Where:** Agent localisation, a system has some sense of location of the agents that interact with it, either geographically or within predetermined bounds such as a map or floor plan.
- **What:** Activity Recognition, a system has the means to determine the activities of the agents; the activities will often be drawn from a predetermined set of activities based on the system use-case.
- **Who:** Identity Awareness, a system is able to identify the main agents that interact with it and determine the significance of that interaction based on the identification.
- **Why:** As stated in (Brooks, 2003), this is perhaps the most difficult element of a system to determine and refers to the ability of a system to understand intention, however here it will be defined as the underlying intention for the system by the designers i.e. its purpose (why) in recognising a particular activity (what), at a particular time (when), by an agent (who) in an ambient environment (where).

This thesis is primarily concerned with the “what” and the “when” of the context quintet, but these elements do not provide context without some input from the remaining quintet and indeed the scope is significantly narrowed by the thesis aim. In general the aim of this research is to investigate, develop and evaluate a system for non-intrusive monitoring within an ambient environment (the “where”) occupied by a single elderly occupant (the “who”), to recognise the activities (the “what”), and hence offer a system solution in the area of assisted living (the “why”), that is more practically deployable than systems that are currently proposed. Specifically the research seeks to investigate the use of Ubiquitous sensors within a smart home setting with a view to establishing whether activity recognition is possible with a reduced, less intrusive subset of sensors that can be realised using utility meter disaggregation techniques. The research considers the selection of sensors as a feature selection problem and with an established method of activity recognition that implements a reduced number of sensors it can be argued that occupants of the monitored environment maintain a greater level of privacy.

1.2 Challenges in the field of Activity Recognition

Recognising human activity from data produced by sensors in an ambient environment has many challenges. These challenges manifest in the ability to train HAR systems on quality labelled training data that is relevant to the sensors in the environment and also the complexity and stochastic nature of human activity. For HAR systems to be useful in practice they will need to be deployed in real environments with minimum disruption to the occupants while maintaining an ability to respond effectively to useful data produced by that environment. These systems should require the minimum of system training, configuration and tuning and should be flexible enough to adapt to the occupants and moreover, accepted as usable by those occupants. What follows is a more detailed discussion of some of these challenges:

1.2.1 Insufficient labelled data

HAR systems use classification of data from sensors to detect activities within the assisted living environment. Although machine learning is the most common technique in the literature, and will be discussed in detail in the chapters that follow, rule based systems have also been proposed with some degree of success (Bae, 2014), (Storf, et al., 2009). Rule based systems require domain knowledge to establish the rules and so suffer from the same period of pre-monitoring that is a requirement of machine learning systems and so are not considered here.

In order to train the models used for classification with machine learning, there is a requirement for reasonable quantities of accurately labelled training data that reflect the activities of the environment that it is used in. This labelled training data may be required at each level of human motion, as described in chapter 2, for example an HAR system using a video camera may need labelled training data to determine if the current series of frames represents an event caused by the movement of the occupant turning on the cooker. At the next level, labelled training data is required to learn a generalisation of the activity instances that represent the activities that are the focus of the HAR system in question, for example the various sequences of previous level events that signify the activity of “cooking” will need to be learned from numerous labelled sequences of such an activity taking place, and at the next level there may be a requirement for the HAR

system to model habitual patterns of behaviour, and so may require labelled training data to determine the patterns that signify the cooking of breakfast, Lunch and dinner over a weekly cycle lasting a year. In laboratory environments the most common methods of collecting labelled training data use hand labelling techniques (Tapia, et al., 2004), (Crandall & Cook, 2008), (Fishkin, et al., 2005). This is not a practical approach for useful deployment in an assisted living environment as it indicates a need for long term annotation of data by a combination of the commissioning engineer and the resident. Any deployed system that is reliant on data that is labelled by the resident, even if the resident is meticulous and the annotation is highly accurate, will suffer from an imbalance in labelled data. That is to say that; those activities that are rare will be underrepresented in the learned model (He & Garcia, 2009). The system proposed here uses a novel method of data synthesis that alleviates the need for long periods of data capture and labelling by using synthesis of labelled training data at both utility meter and at the activity recognition stage.

1.2.2 Intra and inter-class variations

When classifying human activity, challenges manifest when activity instances of the same class can have sensor signals that appear in a different order in the sequence, appear a varying number of times or do not appear at all in some instances but do in others. This is referred to as Intra-class variations and was studied by (Rashidi & Cook, 2009) in the context of HAR, and is also evident when sensors deployed from environment to environment have different viewpoints or variations in the way the occupants carry out the same activity in different ways. Problems also exist when differentiating between instances of different activity classes; these are referred to as Inter-class variations and were also studied by (Rashidi & Cook, 2009), and are defined by a lack of variation between sensor sequences of different activity classes. Such similar sequences can often confuse a machine learning algorithm, particularly if it is trained with limited labelled data and the activity is represented by a large number of features (Friedman, 1997), ultimately resulting in the misclassification of the activity. Ideally an HAR system should be designed to minimise Intra-class variations and maximise Inter-class variations.

1.2.3 Multiple Occupancy

When an environment is occupied by more than one person the sophistication of the HAR learning system will need to increase exponentially for each extra occupant, if each occupant's event sequence for each activity differs significantly (Crandall & Cook, 2009). There are three aspects to this particular challenge, one of which relates directly to Intra class variations, where each occupant may perform the same activity using different variations in the sequence of actions that define that activity, thus suggesting that the HAR system must be trained on each activity for each occupant. Another aspect to this challenge is that of collaborative engagement in an activity where multiple occupants cooperate to share the actions that make up an activity sequence. The final aspect is when an environment has multiple active occupants, where the actions of each occupant and hence the movement events sensed for each occupant will occur in parallel and so these actions become interleaved, the challenge here is to associate each sensor signal with the action of the occupant that triggered it.

1.2.4 Privacy

Privacy is a common concern and remains a challenge to the practical implementation of HAR systems. Privacy is defined by the Oxford English Dictionary (Anon., 2007) as "The state or condition of being alone, undisturbed, or free from attention". Clearly the very aim of HAR systems is to recognise and monitor the actions of people in an ambient environment, so they are inherently designed to observe and hence infringe on the privacy of the occupants. In a review of future applications in dementia care by (Bharucha, et al., 2009) the authors state that 'the very systems that are designed to promote independence require varying degrees of privacy impingements'.

There are several means for collecting movement data within HAR environments that can be arranged into two main categories; vision-based systems and sensor-based systems (Chen, et al., 2012). Out of these two categories, vision-based systems raise the most concerns over the violation of the occupant's privacy (Magnusson & Hanson, 2003). This is especially the case if the occupants have no control over the video devices. This does not mean that sensor-based

systems, with no visual element, are accepted entirely but there seems to be a granularity in the type of movement data that is perceived as either highly sensitive or generic (Garg, et al., 2014). In the research conducted by (Garg, et al., 2014) the results suggest that the 101 elderly residents were reluctant to share highly sensitive information unless there was an explicit utility for sharing this information, the example provided is when the activity data is used to indicate an alert if a resident falls. The study concludes that very specific and explicit needs, fulfilled by HAR systems, could well outweigh the privacy concerns indicated.

1.3 Thesis structure and contributions

This research makes contributions in the following areas:

- A significant contribution is made and the thesis hypothesis is validated by showing comparable activity recognition using a reduced sensor set that is represented by movement, electrical appliance and water usage. A novel feature sub selection algorithm is demonstrated that uses embedded feature selection followed by wrapper sub selection using backward elimination. This contribution is in the process of adaptation for publication.
- A published contribution was made (Wonders, et al., 2016) which demonstrates a technique for synthesis of labelled training data for use with machine learning algorithms when disaggregating at the water meter.
- A significant contribution is made in the development of an environment using disaggregation at the utility meters and motion sensors to infer the activities of the inhabitant. Published work on such a system of activity recognition is not available at the time of writing and the work is in the process of adaptation for publication.

A summary of the contents of each chapter follows highlighting the areas where the contributions are discussed.

Chapter 2: Background Information and Literature Review

Chapter 2 introduces the definitions that underpin the fundamental concepts in the field of Human Activity Recognition. A three tier hierarchical representation of human activity is suggested using relevant literature from the field and a context has been provided that relates these three tiers to computational classification of human behaviour. The challenges in activity recognition, as defined in section 1.2, are highlighted within a review of common approaches to human activity recognition and separated into two distinct fields that utilise either body worn sensors or environmental sensors. The chapter concludes by defining the computational representation of movement events, activities and behaviour for use by activity recognition algorithms.

Chapter 3: Machine learning for Human Activity Recognition

Chapter 3 discusses the privacy challenge faced for Human Activity Recognition Systems and positions this challenge in the context of HAR in ambient environments, indicating why activity recognition using binary sensors may be a more acceptable and practical method for use in assisted living applications. The underpinning principles of Machine Learning are introduced as the dominant technology in the field with supervised learning and classification being highlighted as the dominant techniques. Examples of how supervised algorithms have been used and their performance for activity recognition to date are provided. A Machine Learning analysis framework is introduced for use throughout the remainder of the thesis and data pre-processing is discussed with a view to justifying the use of simple features that relate transparently to the sensors of interest, that relate directly to the key aims of the thesis, as provided in section 1.1. The chapter introduces two publicly available datasets as generated in studies by (Tapia, et al., 2004) and (Crandall & Cook, 2008) chosen as they go some way to highlighting some of the challenges in the field relating to the quantity of labelled training data and the stochastic and unbalanced nature of human activity data. The metrics used for evaluation of classification results are discussed in light of these challenges with more enlightening metrics being suggested to provide more realistic comparisons of algorithm performance.

Chapter 4: Feature Selection for activity recognition

Chapter 4 describes the analysis of two data sets that was carried out with a view to determine those sensors that contribute most to the performance of Machine Learning classification models that perform well on the publically available, Human Activity, datasets used. The concept of feature selection is discussed and definitions of three common feature selection methods are introduced. These methods are reviewed in terms of appropriate use for sensor selection and a novel hybrid feature selection method, that suits the purposes of Activity recognition in ambient environments, is proposed. Six classification algorithms are evaluated using appropriate tools and on the selected, publicly available datasets with the significance of the results discussed. The performance of the proposed feature selection strategy is evaluated and appropriate visualisations produced throughout. The chapter is summarised with a discussion of the results and confirming that, using the (Tapia, et al., 2004) dataset the number of sensors can be reduced from 72 to 43 with no reduction in classifier performance. This chapter forms a significant contribution as it validates the thesis hypothesis with comparable activity recognition using a reduced sensor set that is represented by movement, electrical appliance and water usage. This work is in the process of adaptation for publication.

Chapter 5: Repurposed Smart Meters

Chapter 5 proposes and demonstrates a method of water disaggregation at the meter point using a purpose built water meter and compares the classification accuracy of Artificial Neural Network (ANN), Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classifiers on predicting the fixture responsible for an event. A method of disaggregation using both labelled data, collected over a two month period, and synthesised data generated from only two extreme examples per class are evaluated. The chapter is a significant contribution to the field of activity recognition as for such HAR systems to be practical, the implementation and training times of the upstream disaggregation system must also be practical. The chapter demonstrates that by using synthesised data, automatically generated by algorithms developed from knowledge of the domain, that training times for water meter disaggregation can be reduced to minutes rather than hours or days. An accuracy of 84.97%, is comparable to that of (Chen, et al., 2005), (Fogarty, et

al., 2006), (Froehlich, et al., 2009), (Srinivasan, et al., 2011), (Larson, et al., 2012), (Nguyen, et al., 2013a), and (Nguyen, et al., 2013b), however these systems rely on real data for training and calibration resulting in significant limitations in terms of time and access to occupants of the environment under study.

Chapter 6: HAR using Sensor Fusion

Chapter 6 describes the development of an ambient environment and the experimentation and evaluation of a Human Activity recognition system that implements the fusion of utility meter disaggregation and movement sensor data for activity classification. Using supervised classification algorithms and single occupancy activity data, collected over a period of 28 days, a base line evaluation of classification performance is documented using “bag of words” sensor counts as the features. These baseline performance results are then compared to those achieved when a temporal feature is added to the feature vector. Both baseline classification results and those obtained with the addition of temporal data are achieved using ground truth labels for water meter disaggregation, therefore the evaluation is repeated using labels provided through classified water disaggregation and compared with the baseline. The chapter concludes with the presentation and demonstration of the effect of using data synthesis of training data at the activity level to balance unbalanced datasets and so avoid training bias toward majority activity labels. This chapter forms a significant contribution as published work on such a system of activity recognition that implements fusion of disaggregation at the utility meters is not available at the time of writing. The work is in the process of adaptation for publication.

Chapter 2 Background

This chapter outlines the underpinning definitions from the field of activity recognition as used in the relevant research literature. In order to establish a common language for discussion, the chapter begins by defining three levels of human motion as used in the context of the research aims and identifies relevant activity classes based on the literature in the field. The main challenges in the field of activity recognition, as described in Chapter 1, are further pointed out in a literature review and related to the aims of the research. Common approaches to Activity recognition are discussed with a view to the evaluation of how appropriate they are to the fulfilment of the research aims. The chapter concludes by defining the data representation of sensor output for computational detection of activities and behaviour.

2.1 Definitions used in HAR

Within the field of HAR and ambient environments several terms are used extensively to describe the hierarchical characterisation of human activity, the definition of these terms varies depending on the sensors and the environment under discussion. The work by (Bobick, 1997) is prominent in the field for providing some early work on human activity, and provides useful definitions based on the concept of the machine perception of human motion where it is hierarchically categorised by movements, activities, and actions. Here the definitions provided by (Bobick, 1997) are used with the exception of the term “behaviour”, the definition of which is taken from the work carried out by (Hull, 1943). A definitive description of each level of human motion is provided and followed up with a definition for the classes of activity that are commonly used in the field.

Level 1 - Human Movement: At the atomic level of human motion lies movement, this motion is characterised by body movements with no temporal or contextual element. Movements are defined by (Bobick, 1997) as being consistent, in terms of execution, from one instance to the next. In video based HAR “the pixel-based description of the motion under different possible viewing conditions is the only knowledge required to see the movement” (Bobick, 1997), for

example the motion required to “turn on a tap” may have little variation from one person to the next. In an environment that implements ubiquitous sensors the simple actuation of a sensor is the only knowledge required to see the movement, for example a sensor designed to detect the turning on of a tap detects the same motion from one individual to the next.

Level 2 - Human Activity: The next level of human motion is the activity; (Bobick, 1997) defines this as a combination of “statistical sequences of movements” and states where “The motion is no longer a single, primitive, consistent movement”. Instances of the same activity may have sequences of movements that are ordered differently or there may be more or less movements in the activity across instances. There is also a temporal element to an activity that may have variations between, and during movements across instances. The instance variations across an activity introduce uncertainty that may or may not be modelled statistically, depending on the activity in question. Following from the example provided in the definition of a movement; the activity of “bathing” is a sequence of movements that can have numerous instance variations in terms of sequence and time. The “bathing” activity will probably involve the “turn on a tap” movement but may include one or two taps in a different order and over varying durations of time. This example further illustrates that seeing a sequence of movements can provide context to the movement in question, for example a sequence of video frames could be used to detect the “turn on tap” movement just after detecting the “enter bathroom” movement followed by the “insert bath plug” movement. In this case the simple atomic movement in question, “turn on tap” is given context by the movements before and after.

Level 3 - Human Behaviour (Action): In the definitions used by (Bobick, 1997) the highest level human motion is described as an action, the description of which seems to overlap that of activity with the addition of high level causal reasoning. For the purposes of this thesis a more intuitive notion of human motion, that includes causal reasoning and encapsulates a sequence of activities, is provided by (Hull, 1943). In the book, *Principles of Behaviour*, Hull describes animals as “aggregations of need” and the behaviour required to alleviate a particular need is conditioned on the environment at the time of need. (Hull, 1943) Further defines behaviour as an

“innate molar response” that consists of a sequence of actions, often formed from habit. This particular high level definition of movements suits the purposes of this thesis well, as when the conditions within an environment developed for assisted living can be controlled, (Hull, 1943) suggests that human behaviour, brought on by a specific need, in such an environment may be deterministic and thus could be generalizable and classified. Note that here we substitute the term “action” in the definition provided by (Hull, 1943), with the previously defined term; activity. An example of behaviour as a sequence of activities formed from habit and involving causal reasoning could be the behaviour of taking a bath on a Friday before dinner.

2.2 Classes of Activity

Much of the research carried out in HAR requires definitions and terms based around the activities being detected by that system, the sensors used, the applied models, and the ultimate aim of the system. As a result there is commonly a divergence of disparate terminology, labelling and structure of data across the research field and It follows that not all of the research can be generalised for comparison with past, current and future work. The terms used in this field do vary somewhat, but are generally drawn from the work produced in (Lawton & Brody, 1988) where the Instrumental Activities of Daily Living (IADL) was defined for use in clinical practice as part of a comprehensive assessment to determine the functional ability of elderly patients in the home. The IADL scale defines 8 categories of functional measurement; Ability to use the telephone, Shopping, Food preparation, Housekeeping, Laundry, Mode of Transportation, Responsibility for Own Medications and ability to handle finances, each of these domains include a number of assessment activities. It can be seen from the list of activities and domains defined in this scale that some of the general definitions may be difficult to measure with a sensor based environment, and it may be difficult to ethically determine an inhabitant’s ability to handle finances using current HAR techniques. These types of scales are also designed for use as an assessment tool with a requirement for a skilled assessor to make judgements regarding the performance of a task by the patient/inhabitant and so often do not translate well to the automated world of software and sensors. In light of the difficulties in adapting these scales to the recognition of activities using computational methods with limited sensor capabilities, this work

will propose a method of pre-monitoring of an environment to determine the activities that are carried out during the normal daily routine of a resident. This is justifiable as there are no current agreed activity set definitions for comparison across studies and it should be noted that the views of the researcher are reflected in the work by (Krishnan, et al., 2013) , where it is argued that in order for significant progress to be made in human activity recognition research, there is a need for standardisation of the definitions used so that studies can be repeated and research data scaled and adapted across disparate environments.

2.3 Approaches to HAR

The problem of HAR can be defined as the classification of the computational perception of movement data produced by sensing devices that are focused on that human movement. That is, the aim of an HAR system is to map the raw data provided by the sensing devices, either vision-based, sensor-based or both, onto a class label taken from a set of activity labels. This section provides an overview of some of the literature in the field of HAR with a focus on the challenges mentioned in chapter 1. The literature in the field of HAR points to two main areas for the location of sensors; Sensors that are located on the body and used to determine the physical activity of the wearer as described in (Aminian, et al., 1999), (Najafi, et al., 2003), (Parkka, et al., 2006), (Atallah, et al., 2011), (Liu, et al., 2012) and (Anjum & Ilyas, 2013) as well as sensors that are located in the environment and are used to detect environmental movement and interaction, (Cook, et al., 2003), (Hayes, et al., 2008), (Cook, et al., 2009), (Augusto, et al., 2010), and (Lee, et al., 2013). It should be stressed at this point that the review of physical activity using body worn sensors is included to provide an overall background to the thesis, and to provide context and continuity in terms of the challenges faced in the field; however the aims of the thesis remain in the area of activity recognition using environmental sensors. In the category of vision-based systems, video cameras have been used to provide a combination of physical activity recognition data (Brand & Kettner, 2000), (Ke, et al., 2013) and environmental movement and interaction data (Nguyen, et al., 2003), (Ayers & Shah, 2001), however the focus of this thesis is in part based around tackling the challenge of privacy using

activity recognition systems and so sensor based systems are preferred to those using video data for activity classification.

2.3.1 Recognition of physical activity

Physical activity was defined by (Caspersen, et al., 1985) as; “*any bodily movement produced by skeletal muscles that results in energy expenditure.*”, the authors went on to categorise physical activity in daily life into occupational activity, sports activity, conditioning activity, household activity or other activity. By using suitable sensing devices that can detect or measure, skeletal muscle movement or energy expenditure there is a potential to detect human physical activity and use this for activity recognition. It should be noted that in the following literature review the terminology used by the authors of the publications is maintained within the review, however with reference to the definitions provided in section 2.1 the first level of human motion; “movement” is interchangeable with the term “activity” used throughout the reviews. This is justified as quite often the aim of physical activity recognition is to classify the particular movement of the subject within the situational context, for example sitting, standing or walking.

An early study carried out by (Aminian, et al., 1999) to determine what was termed “daily physical activities” implemented two accelerometers, one attached to the chest to measure vertical acceleration and another attached to the thigh to measure horizontal acceleration. The study focuses on the activities of lying, sitting, standing and treadmill walking. The reported results indicate a misclassification error of 10.7%. It is interesting to note that the authors state that reducing the number of sensors to include only the chest sensors increases the misclassification error and claim that two sensor sites may provide optimum classification for the chosen activity classes.

In contrast to the suggestions made by (Aminian, et al., 1999) the research produced by (Najafi, et al., 2003) implemented a single chest worn device to detect the physical activities of sitting, standing, lying and walking. The study was carried out in the context of assessment for activities of daily living and behaviour monitoring. The three sensors consisted of two accelerometers which measured the vertical and horizontal velocity of the subject and a gyroscope which

measured the angular velocity through the sagittal plane. The authors report that by using vertical displacement for the detection of transitional movements improved results from previously reported studies, when combined with vertical acceleration data. The research is interesting because of the use of a single body worn device, where the authors claim minimal interference with the usual activities of the subject. The placement of the sensing device is reported as an important issue and the authors claim that the kinematic and gravitational outputs of the accelerometer and gyroscope, in this study, were dependent on a chest mounted device, but acknowledge the need for further research into sensor count and location.

(Parkka, et al., 2006) Picks up on the issue of sensor count and location and widens the scope by including sensor type, and researches the use of a range of sensors to classify the physical activities of lying, sitting/standing, walking, Nordic walking, running, rowing and cycling. It is interesting to see that although the use of many variable sensor types has the potential to provide an abundance of features to aid in the classification process, the features chosen were those taken from devices that were located on moving parts of the body rather than those monitoring the physical health of the wearer. The authors mention that the absence of accelerometers on the lower body was a limitation of the study, but again, backing up the conclusions of (Najafi, et al., 2003) indicate that placement of these sensors should be considered carefully.

This claim is investigated by the work carried out by (Atallah, et al., 2011) on the analysis of sensor placement and feature ranking for physical activity recognition.

Table 2.1: *List of activities and group classifications (Atallah, et al., 2011)*

Activity group	Activity
Very low level activity	1. Lying down
Low level activity	2. Preparing Food 3. Eating and Drinking 4. Socialising 5. Reading 6. Getting Dressed
Medium level activity	7. Walking in a corridor 8. Treadmill walking at 2 km/h 9. Vacuuming 10. Wiping tables
High level activity	11. Running in a corridor 12. Treadmill running at 7 km/h 13. Cycling
Transitional activity	14. Sitting down and getting up (5 repetitions) 15. Lying down and getting up (5 repetitions)

In this work the researcher's group fifteen activities of daily living into five categories based on the compendium of physical activities (Ainsworth, et al., 2000). The researchers placed seven, 3-axis accelerometers to the ear, chest, arm, wrist, waist, knee and ankle of the subjects to determine the sensors that had most impact on the classification of the activities listed in Table 2.1. The researchers argue that consistent and practical placement of sensors plays an important role in the accurate classification of low-level activities, as listed in Table 2.1. Consistency of sensor placement is problematic because the majority of studies implement sensors to record data for very specific areas of movement; (Najafi, et al., 2003), (Atallah, et al., 2011). Clearly any practical system of activity recognition should be capable of generic classification despite slight variations in sensor position in a specific area. In conclusion the study backs up the claim made by (Parkka, et al., 2006) that the addition of lower body sensors increases the ability of classifiers to detect physical activities. More importantly the research provides an important insight into the relevance of sensor position as it relates to the activity of interest.

All of the approaches to physical activity recognition mentioned so far include accelerometers as either one in a range of sensors types or the only sensor type used to determine the activity of the subject. It is not surprising then, that with the inclusion of accelerometer sensors within the modern smart phone; researchers have conducted studies on the use of the mobile phone for recognition of physical activities. Although from the reviews discussed so far the key issues have been the number and location of sensors, it seems to be a deviation to suggest that a single sensor in an unknown location (unknown because the wearer is at liberty to carry a mobile phone in any location) can improve on recognition accuracy over that of several sensors measuring various movement locations. The suggestion made in the review of (Liu, et al., 2012) that by using more computationally expensive and adaptable classification techniques such as the SVM classifier it could be possible to create more general classifiers that can cope with variations in sensor location and across sensor subjects. Research carried out by (Anjum & Ilyas, 2013) discusses the issues of practicality when implementing numerous body worn sensors for classification of physical activity and as such developed a mobile phone application to study the use of the built in accelerometer, gyroscope and Global Positioning System (GPS). The aim was to classify the

daily activities of walking, running, climbing stairs, descending stairs, cycling, driving and inactivity. The study acknowledges the need for large labelled data sets to create effective classification models that use supervised learning algorithms; this is acknowledged and is highlighted in each of the studies that implement supervised learning algorithms reviewed to this point, the technical details of this particular issue are deferred until discussions in Chapter 3 and 4 of this thesis but it is noted here as a current challenge. The study reports the use of a relatively balanced data set; i.e. there are a similar number of labelled data points across each activity and the number of data points collected for each mobile phone position is within a reasonable range, again this point will be discussed in further detail in chapters 3, 4, 5 and 6. The results reported are comparable with previous studies and is a promising view into what could be possible when the number of sensors used are restricted to those available from a mobile phone and highlights the requirement for analysis of those sensors that contribute to the accurate recognition of the activity of interest.

2.3.2 Activity recognition using environmental sensors

As well as using sensors that are worn by the subject whose activities are the focus of attention, sensors can also be placed in the environment occupied by the subject, these sensitised environments are often referred to as “ambient intelligent environments” (AmI) as discussed in (Sanchez, et al., 2007). Such an environment is useful when the challenges faced by attaching sensors to the body are considered and the context of the activity of focus is within that environment. The advantage of sensors distributed in a residential environment is that the sensor location is fixed, as the particular subject of focus moves within the environment. It should be noted, however that data produced for one environment may not be relevant to another, as sensors are often attached in varying locations and environments have various configurations. As recognised by (Atallah, et al., 2011), when using multiple body worn sensors to detect the activity of a subject there is a desire, often with the aim of minimising the number of sensors required, to determine which sensors provide data that is relevant to the recognition of that activity, this issue is no different and indeed more pronounced when multiple sensors, often many more than used for activity recognition with body worn sensors, are distributed within an ambient

environment. The reviewed literature highlights several research projects that have developed ambient environments with significant contributions to the field of human activity recognition, and also investigate and highlight the challenges mentioned in chapter 1.

Gator Tech Smart House - University of Florida: The Mobile and Pervasive Computing Laboratory (MPCL) at the University of Florida created the Gator Tech Smart House in 2005 (Helal, et al., 2005) to address the adaptability limitations of what is termed “pervasive computing systems” as new technologies emerge. The environment is a fully functional house with a variety of sensor types ranging from Radio Frequency Identification (RFID) sensors, microphones, video, contact sensors, motion sensors and a smart floor mapping systems that implements pressure sensors on tiles to form a location sensitive pressure grid. There are several interesting publications relating to the use of the Gator tech environment for activity recognition. An example that seems unique to the Gator Tech environment, is an early study carried out by (Bose & Helal, 2008) which focused on the monitoring of walking patterns using the pressure sensor network embedded in the floor of the environment. The research describes the floor as consisting of a grid of piezoelectric force sensors embedded under the floor tiles of a 2,500 square foot floor. These sensors are used to track the motion of a subject within the environment. The study claims that this technique is capable of measuring the three characteristics of walking motion; stride length, gait velocity, and cadence without the need for encumbering body worn sensors, and that these measurements can be derived from simple mathematical calculations that need no a priori knowledge of the subject and hence, no labelled training data is required. The results of experimentation are not provided, so cannot be verified but the mechanism is an intriguing alternative to more invasive vision-based motion tracking systems and also is a consideration when the aim is to mitigate the issues related to the requirement for large quantities of labelled training data. The challenges relating to multiple occupants could also benefit from this type of footfall tracking system, as often one of the related issues in this challenge is that of assigning sensor events to each individual in the environment (Crandall & Cook, 2008).

In an article published by (Helal, et al., 2009) the Gator tech house was presented as one environment used as a Smart Home-based platform for behavioural monitoring of the residents.

The report outlines a monitoring and analysis platform that combines personal wearables and ambient environment sensors to form a flexible and extensible system for monitoring the health of the occupants. The test beds used for the system are the Gator tech Smart House (Helal, et al., 2005) and the Centre for Advanced Studies in Adaptive Systems (CASAS) smart apartment at Washington State University (Cook, et al., 2009). The CASAS smart apartment is equipped with a grid of motion sensors to detect the movement of subjects throughout the house as well as sensors to detect the use of hot and cold water as well as the use of a stove burner. Telephone usage is captured by voice over IP (VOIP) and contact sensors monitor the subject's interactions with items such as a cooking pot and what is termed "key cooking ingredients" as well as a medicine container. (Helal, et al., 2009) Combine the use of body worn sensors and environmental sensors in the development of a flexible system that can be deployed at the point of need such as a laboratory, a clinical environment or assisted living spaces. The system uses Hidden Markov models (HMMs) trained on each activity to probabilistically infer the most likely activity being carried out by the occupants. HMM's were learned for five activities, looking up and calling a phone number, washing hands, cooking, eating while taking medicine, and washing dishes, and the study reports a classification accuracy of 98% using three-fold cross validation. What is interesting about this study is the reported combined use of body-worn and ambient sensors using flexible plug and play architectures. It's seems that if activity recognition systems are to be adopted in real world environments this type of plug and play concept will be an essential requirement to enable quick, relatively non-technical retrofit integration into existing spaces and thus allowing the occupant to remain in their home and benefit from non-invasive monitoring of activities. In order to develop such flexible plug and play systems, the challenge of collecting sufficient labelled training data remains a significant challenge. (Mendez-Vazquez, et al., 2009) produced research attributed to the MPCL and the Gator Tech environment, on the production of synthetic labelled training data for use in machine learning model training of activity recognition systems. This publication uses Markov Chains (MCs) to generate simulated activity events, and hypothesises that because Markov chains have been successfully implemented to detect activity patterns the reverse process could feasibly be used to produce activities. This is a reasonable hypothesis because an MC, in the context of activity recognition,

uses the probability of transitioning from one movement to the next to determine the probability of a sequence of movements. If each activity is classified by a unique MC then the current activity classification would be the MC, and hence activity, with the most probable sequence. The inverse process, used to generate data for a particular activity, would carry out a random walk, based on the probability transitions, through the MC generating a movement event at each random probability transition, this process can be repeated for each activity and so can be used to generate a large labelled training dataset. In order to capture timestamp information for each event, values are drawn from a Poisson distribution that is created from a priori knowledge. The idea of generating synthesised labelled data sets to aid training of machine learning models for human activity recognition as well as for sharing among the research community is an interesting one and is followed up by the MPCL in (Helal, et al., 2011) where it is suggested that powerful and realistic simulation tools can be used to support and progress research in the field. The research paper presents a software simulator named Persim, designed to synthesise datasets based on human activities. Follow up studies from the MCPL based on the development of Persim include (Helal, et al., 2012) where a virtual three dimensional environment is proposed as the interface to Persim, (Lee, et al., 2013) , (Lee, et al., 2014) , while (Lee, et al., 2015) reports on the development and integration of a context driven algorithm to improve the computational complexity of Persim. Simulation techniques and algorithms provide a convincing mechanism to mitigate the on-going challenge of insufficient labelled training data and as a side effect provide a means of generating large datasets for sharing between the research communities. This thesis contributes in the area of labelled training data syntheses at the level of movement detection where data is simulated for disaggregation at the water meter and for labelled training data to train an activity recognition system.

CASAS smart apartment - Washington State University: The CASAS project lists one of the active areas of research as “Activity Learning - Discovery, Recognition, and Prediction” and has numerous publications of interest. A notable example is (Cook, et al., 2009) where the focus of the publication is on collecting and disseminating sensor data from the CASAS environment where the researchers argue that “shared home behaviour datasets are critical in order to test,

compare and enhance smart home user modelling and activity recognition”. The initial sensor data collected was based on five activities; Telephone use, Hand washing, Meal Preparation, Eating and Medication Use (Note: this was a combined activity) and Cleaning, all selected from the IADL scale discussed in section 2.1. The format of the data collected from the sensors is as defined in section 2.3.3, and includes a timestamp, a label representing the ID of the sensor and a message relating to the state of the sensor. The initial dataset was generated from subjects performing the activities of interest, while events were continuously recorded, no time scale is provided but the study resulted in a dataset of 5,312 sensor events. The publication discusses several datasets that were collected during experimentation some of which relate to the challenges of activity recognition discussed in chapter 1. Datasets that represent incomplete or Erroneous activities, Interweaved activities (these are activities that start when another is not yet complete etc.) and Multiple Resident Activity Data. Interestingly the researchers have documented the challenges that were faced when collecting activity recognition data as:

- 1) Clean Data: The difficulty in collecting clean data and state failures in sensors and equipment as dramatic setbacks in the data collection process.
- 2) Data Annotation: The labelling of data sequences with the activity class that the sequence represents.
- 3) Generating sufficiently varied data: The datasets collected cannot be guaranteed to have captured all possible variations that would be encountered in a natural environment; as such the algorithms used for classification must be robust to these variations which are a much greater issue when trying to create generic data sets that can be used across environments.

It is an interesting observation that the challenges of labelled data collection and dissemination exist whether the sensors are placed on the body of the subject or in the environment used by the subject. It is also an interesting observation that the article was published in 2009 and still the challenge of insufficient quantities of standardised labelled training data has not been addressed adequately. An earlier study listed as part of the CASAS project, but implementing activity recognition within an environment referred to as the “Managing An intelligent Versatile” (MAV) Home (Cook, et al., 2003), discusses the use of the temporal properties of activities to help

determine and classify those activities. Note: It is assumed that the MAV Home project was the precursor to the CASAS project based on the identity of the principle author of key publications. The study suggests that by using time intervals as temporal relations such as; before, after, overlaps, start, overlapped-by, finishes and so on, the activities in an ambient environment that bare these temporal relationships can be contextualised using temporal logic. The example activity provided is that of “watching television (TV)” where a subject performs the movement of turning on the TV after the movement of sitting on the couch, the indication here is that these movements are related with the temporal relation “after” therefore it is more probable that in the future if the sitting on the couch movement is detected then the turning on the TV movement will follow. The publication describes the use of a rules based temporal algorithm that is used to learn the temporal relationships between activities and was trained on 59 days of data and tested on 1 day of data. The researchers report an improvement on sequential rule based algorithms of 1.86% classification accuracy; however this is an early study and is included as indication that temporal properties could possibly play a major role in the development of activity recognition systems in ambient environments, a point which is tested in the final chapter of this thesis. Another article of interest attributed to the CASAS project (Crandall & Cook, 2008) has a focus on the challenge of activity recognition in an environment with multiple occupants, more specifically the issues related to attributing sensor events to each occupant in an ambient environment. The research used data generated by three occupants of a sensitised office space that was monitored for several weeks producing a dataset containing 6000 unique events. The events were generated by a combination of motion sensors providing complete coverage of the space, door reed switches monitoring the opening and closing of the entrance and exit to the office and monitored light switches. The data was labelled by the occupants as they used the environment so that each event was attributed to the occupant that generated it, these events were then filtered to only include those generated when each individual was alone in the environment, with the aim of building an activity pattern for each occupant. Several Naïve Bayes (NB) classifiers were trained and tested using features based on various temporal cycles, these temporal classifiers were compared with a baseline NB classifier trained with the timestamp and no other temporal feature. The temporal features implemented were “Hour of Day” , “Day of Week”, “Part of Week” (a categorical

variable taking values of Weekday or Weekend), and “Part of Day” (a categorical variable taking values of Morning, Afternoon and Evening). The researchers report an issue with imbalanced data sets where one individual (referred to with the Pseudonym John) was responsible for 62% of the data events and so attributing all events to this individual would yield an accuracy of 62%, however false positives would also be high at 48%. The issue of imbalanced datasets relates directly to the challenge of insufficient labelled training data, as data is difficult and time consuming to collect; those events that occur frequently tend to dominate the dataset while infrequent events are underrepresented. A solution to this issue is described and tested in the final chapter of this thesis. When using no temporal data the researchers report over 90% accuracy and false positives above 30% for the occupant generating the majority of events. The other occupants (referred to with the Pseudonyms Abe and Charlie) have reported accuracies of 70% and 30% and false positives of less than 5%, however when comparing these results to those taken using each of the temporal features it is interesting to note that the study reports “Hour of Day” as the only feature with a significant positive contribution to classifier accuracy. When the hourly temporal cycle is used as a feature, the accuracy for classifying the events associated with occupant “John” remains very high at over 90% but the false positives are reduced to less than 10% and the accuracy for the other occupant’s increases. For example the accuracy for occupant “Charlie” increases to just less than 90% with false positives just less than 10%. These results go some way to backing up the importance of temporal data in activity recognition in ambient environments claimed in (Cook, et al., 2003), a claim that is again applied and tested in the final chapter of this thesis. Another point of note in the research, reported by (Crandall & Cook, 2008), is an investigation into the relative importance of each of the motion sensors to the differentiation between occupants. The Hypothesis described is that of short lived sequences of motion events being more deterministic of an individual moving throughout the environment rather than an activity of interest in a specific location. Using this hypothesis the researchers filtered the data using a threshold to remove all events with very short durations and retrained the models, resulting in a further improvement to 95% accuracy and 2% false positives using the “Hour of Day” model. This is interesting because it suggests that some sensors may hinder the accurate classification of activities. This also relates to the challenge of privacy in that the higher the data

definition the more invasive the approach, with more sensors the data reveals more detail on the subject as well as being more invasive to fit and maintain. The publication describes a situation where by analysing the relevance of particular sensors to the classification problem, fewer sensors may be used to achieve equivalent results, highlighting one of the core aims of this thesis and a subject which will be taken up in chapters 3 and 4. The work carried out by the CASAS research group also picks up on the challenge of insufficient labelled training data with research carried out by (Szewczyk, et al., 2009) where the effectiveness of four different methods of annotation of environmental data were investigated;

- 1) Raw data annotated after collection by human annotators inferring the activity based on sensor sequences and the time of day,
- 2) Raw data supplemented with resident diaries and inferred by human annotators,
- 3) The use of a visualisation tool referred to as CASASim that is designed to playback events from sensors in real time and thus were inferred by human annotators based on this playback sequence.
- 4) The uses of CASASim play back and supplemented with resident diaries and inferred by human annotators.

The activities of focus in the annotation process were chosen as sleeping, eating, personal Hygiene, preparing a meal, working at a Computer, Watching TV and Other. The study used each of the annotated data sets to train a NB classier using a fixed time sliding window to select an event sequence for recognition. The research concludes that the use of resident feedback significantly increases accuracy and decreases the annotation time but is invasive in terms of the input required by the occupant. The use of a visualisation tool significantly increases the accuracy of the model as annotators are able to play back and visualise the sequence of events in real time. This is significant because by using a software tool to play back event sequences, annotation can be accomplished remotely and so provides a less invasive method of collecting labelled data. Such a tool is also significant when considering the challenge of insufficient

labelled data, as the less invasive the technique for collecting data the greater opportunity to collect large quantities of such data.

The CASAS group have also tackled the challenge of Intra and inter-class variations (Rashidi & Cook, 2009) and the difficulty of creating models that can adapt to the variability in activities carried out across subjects is discussed. These variations manifest when the same activity is carried out differently across different subjects, the term given in (Rashidi & Cook, 2009) is “inter-subject variability” which is equivalent to intra-class variations across subjects, as the activity with the same class has some variability in the event sequence across instances. The term “intra-subject variability” is used to describe intra-class variations when the same subject carries out an activity in different ways across instances of that activity. The researchers describe a model based on the idea of “Transfer Learning” where labelled data collected from a subject is used to learn activity recognition models for a different subject. The system was tested using 23 participants who were asked to follow a script of five activities; telephone usage, hand washing, meal preparation, medication use and household cleaning. Noise, in the form of random events was injected between activities to test the ability of the approach as noise is increased. The researchers report that activities were mapped with a 100% accuracy at noise levels below 20% for long and short sequences. When the noise levels were raised above this threshold the system performs steadily at 85% accuracy but for long sequences the accuracy drops from 80% with 30% noise to 60% when 70% noise is added.

The idea of transfer learning was also used by (Rashidi & Cook, 2010) in an attempt to mitigate the challenge of insufficient labelled data as well as provide a mechanism to build on datasets produced by other researchers in the field. The researchers propose the idea of Multi Home Transfer Learning (MHTL). MHTL is based around the ideas discussed in (Rashidi & Cook, 2009), but instead of using data from one occupant to train models for another the data here is taken from learned activities in one environment and is used as a basis for learning models for another physical space, which may have different physical attributes and sensors. The study assumes that each sensor event is represented with a labelled tuple; $\langle \mathbf{ts}, \mathbf{s}, \mathbf{l} \rangle$ where \mathbf{ts} is the Timestamp, \mathbf{s} is the sensor ID and \mathbf{l} represents the activity label assigned to the event. An activity

is defined as the labelled tuple $\langle \epsilon, \mathbf{l}, \mathbf{t}, \mathbf{d}, \mathbf{L} \rangle$ where ϵ represents a sequence of sensor events, \mathbf{l} is the activity label, \mathbf{t} and \mathbf{d} are the start time and the duration of the activity and \mathbf{L} are a set of location tags where the activity has occurred. The technique depends on what is described as a semi-Expectation Maximisation (semi-EM) algorithm to determine a mapping of activities from the source environment to the target environment. The algorithm was evaluated on data collected from five smart apartments over a period of three months for three of the apartments and two months for the remaining two. The apartments each had different layouts and all had motion sensors, some had contact sensors on doors and cabinets, and one had item sensors on items that were targeted for activity recognition. Four apartments were used as the source of labelled data and one as the target in each combination of source and target. The data from the target apartment was mined to determine the activities of interest and it is these activities that were mapped to corresponding activities from the source apartment data. The researchers report that it is possible to recognize activities using no labelled training data, by mapping data from previous activity recognition studies to the target environment. This is regardless of the physical layout and sensor arrangements of the source or target environments, but limited to activities that are shared between environments. This study was followed up in (Feuz & Cook, 2015) where a technique referred to as Feature Space Remapping (FSR) is proposed as a means to use previously learned source knowledge to improve activity recognition performance on target tasks. The idea here is that the original source data is inherently mapped to a specific set of features that relate to the sensors in an environment and FSR is implemented to learn a mapping from this source feature space to a different set of features used in the target, and is commonly referred to as Heterogeneous Transfer Learning (HTL). It is suggested that the FSR technique is not limited to transferring knowledge from a single source environment but there must exist some relationship between each source and the target environment. This is achieved using an ensemble classifier by mapping the target to each source and then training a separate classifier for each of the sources. The output from each of the source classifiers is combined and a voting mechanism used to make a final activity classification. The performance of the FSR technique was evaluated on 18 datasets collected for one month and from different smart apartments, each equipped with motion and doors sensors. The datasets were labelled with 37 activities taken from the IADL scale. The

researchers report that the FSR technique used in the domain of activity recognition shows a best accuracy of just over 60% and acknowledge that although the technique shows promising results in domains such as document classification where the datasets used are feature rich, in the activity recognition domain it is often the case that there are a large number of classes but relatively few features and so it remains an open challenge to improve on the classifier performance when using existing activity recognition datasets in transfer learning.

2.3.3 Sensors and Activity definitions

As discussed in the introduction, an HAR system that uses ubiquitous sensors is composed of a series of sensors located in an environment that has been designed to detect activities based on a particular context. These sensors are used to determine human movement as defined in section 2.1 and it is important for reasons of compatibility and reproducibility that the data generated across the field of research is transferrable, therefore the data representation as it relates to the three levels of human movement is defined here.

Event (Movement): The sensor data and hence the occupant's movement will be defined in this thesis as an event denoted (e). All events (e) generated within the environment will be represented in the form suggested by (Cook, et al., 2009), this publication was dedicated to the collection and dissemination of smart home sensor data and defined an event as a tuple:

$$e = \langle s, t, m \rangle$$

Where t = *Time Stamp*, s = *Sensor ID*, and m = *Sensor Message*. For example a passive infrared sensor (PIR) located in the Main Bedroom triggered at 17:31:18 on 14th August 2015 would produce the following, movement level 1, event:

$$\langle \text{Main_Bedroom}, 2015-08-14\ 17:31:18, \text{on} \rangle$$

Activity: Movements detected by an environment and coded as events are then used by an HAR system to determine the activity of the occupants within that environment. An instance of a particular activity for the purposes of this thesis is defined as a sequence of events of length n :

$$\mathbf{Activity}_1 = \langle e_1 \dots e_n \rangle$$

A particular activity is learned by an HAR system by creating a generalised internal abstraction over all instances of that activity, in order to identify and label that activity in previously unseen data, therefore a learned activity represents a modelled abstraction over all instances of that activity. For example an instance of the activity of bathing may include the following events:

$$\begin{aligned} e_1 &= \langle \text{Bathroom_PIR}, 2015-08-14 17:31:18, \text{ on} \rangle \\ e_2 &= \langle \text{Bathroom_Plug}, 2015-08-14 17:32:01, \text{ on} \rangle \\ e_3 &= \langle \text{Bathroom_Tap}, 2015-08-14 17:32:31, \text{ on} \rangle \\ e_4 &= \langle \text{Bathroom_PIR}, 2015-08-14 17:33:18, \text{ on} \rangle \\ &\dots \\ e_{n-1} &= \langle \text{Bathroom_PIR}, 2015-08-14 18:05:01, \text{ on} \rangle \\ e_n &= \langle \text{Hall_PIR}, 2015-08-14 18:05:48, \text{ on} \rangle \end{aligned}$$

Note: several events have been omitted for clarity. This instance would be represented as a bathing activity instance starting at time t , taken from event e_1 :

$$\mathbf{Bathing}_t = \langle e_1, e_2, e_3, e_4, \dots e_{n-1}, e_n \rangle$$

The activity instance thus takes on the temporal properties of the movement event elements; with the beginning of the activity represented by the time stamp (t) of event 1 (e_1), the end of the activity represented by the time stamp (t) of the last event (e_n), as well as varying time windows between the events within the activity for example in this instance the duration between the time stamp of e_2 and e_1 represents the time between inserting the bath plug and turning on the tap. Note that by extracting the temporal properties activities can be represented with the form:

$$\mathbf{a} = \langle \mathbf{a}, t, s \rangle$$

Where t = Time Stamp, \mathbf{a} = Activity ID, and s = State (on/off).

Behaviour: Activities detected by the HAR system can be monitored over hierarchical time cycles such as minutes, hours, days, weeks, months, and years to determine habitual behaviour of the occupants within the environment. This cyclic, temporal behaviour can be categorised into patterns associated with individuals and thresholds that represent deviations from these patterns. In activity recognition systems, behaviours can be represented by defining activities as states in a transitional sequence with temporal properties that determine the timing of transitions between states. In the simplest form such a state representation can be defined by Finite State Machines (FSM) as shown by (Ayers & Shah, 2001) where FSM's were used to model the behaviour of the occupants in an office environment. The study does have limitations in that the model has no temporal properties and as such has limited use in cyclic behavioural analysis. Many fields where an interest in complex behaviour created from activity states with temporal, sequential relationships have used HMMs to represent the behaviour. A good example of this was shown by (Starner & Pentland, 1995) where HMMs are used to model the sequence of hand gestures in a video image with goal of decoding American Sign Language. The work published by (Brand & Kettner, 2000) also used HMMs and entropy minimisation to organise the activity states that form the behavioural patterns for visualisation and also experimented with the use of the resultant HMMs in anomaly detection. More complex HMM based models have also been developed to model the complexity of behaviour made up from sequential activity patterns, for example (Nguyen, et al., 2003) developed an Abstract Hidden Markov mEmory Model (AHMEM) which is an extension of the Abstract Hidden Markov Model (AHMM) that allows the representation of both state-dependent and context-free behaviours. Hidden Markov Models have also been used to model and play back, through computer simulation, the learned behaviours of the occupants of an environment, (Noury & Hadidi, 2012). Behaviour analysis is beyond the scope of this thesis but the definition is include here to link the definitions of movement provided in section 2.1 to the physical sensor and temporal data.

2.4 Human Activity Recognition Summary

This chapter has introduced some essential definitions that underpin the fundamental concepts in the field of Human Activity Recognition. A three tier hierarchical representation of human activity has been suggested and underpinned by prominent researchers in the field and a context has been provided that relates these three tiers to computational classification of human behaviour. The challenges in activity recognition, as defined in chapter 1, have been highlighted within a review of common approaches to human activity recognition and, for clarity, separated into two distinct fields; activity recognition using body worn sensors to detect the physical activity of the wearer and activity recognition using environmental sensors to detect the interaction of an occupant in a sensitised environment. Although the thesis is focused in the area of activity recognition using environmental sensors, both modalities of activity recognition are reviewed to provide a wider focus on the four challenges discussed and how these challenges span the entire field. It is noteworthy that the research conducted by (Logan, et al., 2007) highlights sensors that detect movement as more suitable to the task than those that sense interactions with objects. The chapter concludes by defining the computational representation of movement events, activities and behaviour for use by activity recognition algorithms.

Chapter 3 Machine Learning for HAR

As discussed in section 1.2 one of the major challengers with the practical implementation of Human Activity Recognition systems within ambient environments is privacy. With the aim of recognising and monitoring the actions of the residents within the environment such systems have the potential to be invasive at several levels. The commissioning phase of these systems can be invasive in terms of the required access to an environment for a period of time; this access is necessary to install equipment, collect labelled training data and to train and test the machine learning algorithms. Once commissioned, and live, such systems may be perceived as infringing on the privacy of the residents by the very nature of the monitoring task, for example (Bharucha, et al., 2009) state that ‘the very systems that are designed to promote independence require varying degrees of privacy impingements’. There are two main modalities for collecting movement data within HAR environments, those based around computer vision (Ke, et al., 2013) and those based around the use of sensors either worn on the body of the residents as demonstrated in (Aminian, et al., 1999), (Atallah, et al., 2011), and (Anjum & Ilyas, 2013) or affixed to the infrastructure of the environment of which (Bose & Helal, 2008), (Crandall & Cook, 2008), and (Feuz & Cook, 2015) are examples. For the purpose of a discussion on privacy, these two modalities can be compared in terms of the sensor granularity or resolution of the data that is collected and used to recognise the activity. Vision-based and Body-worn sensor systems have a high sensor resolution and raise the most concerns over the violation of occupant privacy (Magnusson & Hanson, 2003), whereas low sensor resolution systems, based around simple binary sensors, placed in carefully chosen locations within an environment, may be perceived as producing data of less granularity and thus less sensitive information relating to the activity of the residents. The occupants perception of sensitive information within ambient environments has been explored by (Garg, et al., 2014) where it was found that residents would often offset the degree of information sensitivity against the utility of that information. As the research explains; if the information is used to detect critical activities, such as a fall, then the residents are more acceptant of a more granular system; however the question must be asked as to where this places an ambient environment that is designed to detect, on a continuous basis, general activities of

daily living. A thought experiment that highlights the link between the sensor resolution and privacy violation in an HAR environment, can be carried out by imagining a hypothetical comparison of a video camera in a bedroom used to infer the activity of “sleeping” to that of a pressure sensor in the bed used to determine if the bed is occupied and thus infer the activity of ‘sleeping’. Clearly the vision-based approach provides more opportunity for certainty as to the activities of the residents, but this is at the expense of privacy. In comparison the data provided by the sensor-based approach is ambiguous i.e. the resident may be “Reading” in bed, which would also trigger the pressure sensor, but it could be argued that it is this ambiguity that provides the perception of privacy. The accuracy of the sensor-based activity recognition system can be improved by adding more sensors, such as a sensor that detects if a bedside lamp is on or off, but this in turn increases the resolution and thus the invasiveness of the system. This also raises a further question as to the chosen activities that should be used to determine if an occupant is active and well, within the environment. This chapter introduces the concept of Machine Learning in the field of HAR for ambient environments with a view to forming the basis for identification of those sensors that contribute most to the ability of the machine learning algorithms used to recognise the activities within that environment. The motivation here is to reduce the number of sensors required for an algorithm to recognise the activities of an occupant, while simultaneously increasing the perceived levels of privacy and thus reducing the intrusiveness of the system as a whole. In order to discuss the subject of sensor selection and its relevance to activity recognition, the topic of machine learning is discussed with a view to positioning feature selection into a machine learning analysis framework.

3.1 Machine Learning

The process of Human activity recognition is highly dependent on the field of Machine learning and the techniques and algorithms developed in this field. The recognition of an activity is a classification problem that uses a computational model learned from sensor data to map that data to a label that corresponds to an activity. This definition relates well to the aim of machine learning, which is to use data taken from empirical experimentation to create computer programs that can be used to solve regression and classification problems (Alpaydin, 2014).

In a research paper entitled *Computing Machinery and Intelligence* (Turing, 1950) proposed the question “Can machines think?” and went on to describe a hypothesis where a machine may be designed using an empirical process. Turing suggested that the machine should be thought of as a child and the problem divided into two parts, the programming and the educational process. The educational process being of a cyclical nature whereby experimental teaching of one machine takes place and is evaluated on how well it learns, and then it is compared to another. Turing goes on to describe a punishment based system of teaching a machine, stating that events which shortly precede the occurrence of a punishment signal should be unlikely to be repeated and a reward signal should increase the probability of repetition of an event.

An important feature of Turing’s learning machine is that the complexities of the learning process may mean that the details are largely hidden from those training the machine and that the process of teaching could follow the normal process used to teach a child. The “things” would be pointed out and a label that describes the thing provided to the machine. This early description of how a machine may be made to think has withstood the test of time and what (Turing, 1950) describes as “details that are largely hidden” can be thought of as an unknown target function (Abu-Mostafa, et al., 2012) that represents the underlying relationship between the attributes or features of the “things”. The empirical process of pointing out “things”, by a teacher, is technically referred to as supervised learning (Barlow, 1989).

In the field of machine learning there are generally two main learning types, supervised learning and the alternative unsupervised learning (Barlow, 1989); unsupervised learning is effectively achieved without, to use Turing’s terminology, “things” being named by a teacher. As such the learning takes place with no name or label and so the aim of unsupervised learning is to find structure or patterns in the example data, if the aim is to discover groups or clustered data points this is referred to as clustering (Bishop, 2006), if the aim is to estimate the distribution of data then this is often referred to as density estimation (Alpaydin, 2014).

Machine learning is therefore a method of programming computers in situations when the complexity of the task is too complex for an analytical solution, and examples of data exist so that an empirical solution can be realized by implementing a suitable algorithm that is able to

produce a model that is learned from the data. In order to develop a machine learned model using the supervised approach, the learning algorithm must implement a cost function that relates to Turing's suggestion of a punishment and reward system where a low cost is analogous to reward and a high cost to a punishment. The cost function is therefore related to the difference between the output of the learning algorithm and reality or ground truth, and the aim of the learning process is to minimise this difference across all possible input/output data points. In this chapter the focus of discussion is limited to supervised learning, where a framework and appropriate algorithms are introduced to determine whether the number of sensors required to classify human activity in an ambient environment may be reduced with minimum impact on the accuracy of the machine learned classifier model.

3.1.1 Supervised Learning

Supervised learning is learning that uses examples of data to learn the underlying relationship between the data attributes and a value. A supervised learning algorithm therefore learns a mapping from an input data vector to a target value, inclusive in the data, and hence is provided with a set of input vector to output target pairs; $D = \{(x_i, y_i)_{i=1}^N\}$, here D is known as the data set and N is the number of data points; made up of the input vector x and target value y (Barber, 2006), (Murphy, 2012). Ideally D would be all of the possible data points that have occurred before and will ever occur in the future, often referred to as the sample space (Abu-Mostafa, et al., 2012), unfortunately this is rarely the case and so D is considered to be a reduced sample taken from the sample space. This is an important distinction as if the former were true the goal of learning the input/output mapping would be to achieve 100% accuracy on the training set, this is memorisation of the training data, but when the latter is true a 100% accuracy on the training set does not necessarily indicate a model that represents the true relationship between the inputs and outputs over the sample space. It follows that learning a target function $f: X \rightarrow Y$ that represents the true relationship over the sample space is the goal of the machine learning algorithm, where X is the input space of all possible input vectors and Y is the output space of all possible output values. As the data set used to determine the target function f is considered a sample taken from the sample space, the function learned by the machine learning algorithm

$g: X \rightarrow Y$ is considered an approximation of f and therefore g is one of many possible approximations (theoretically there are many possible samples contributing to the sample space), belonging to a set of hypotheses H . It follows that the learning algorithm is used to choose the function g , from the set of hypothesis H that best approximates the function f . Once the relationship between the training data and the target is learned it can then be used to predict the output for new, previously unseen input vectors, in the hope that g is a good approximation of f .

The attributes of the data are often referred to as features and the output as the target value. The aims of supervised learning algorithms can be subdivided into algorithms that are used for classification of the input feature vector and algorithms that are designed for regression, these aims are differentiated by the properties of the target value. The classification target is a name or label based on categorical values, whereas the target for regression is a real valued attribute. In the field of activity recognition the values produced by sensors are representative of the occupant's movement within the environment, these raw sensor values can be used to represent the features of the data, although often they are transformed in some way but regardless of the final form, in supervised learning it is a vector of these features that is mapped to a label that represents an activity. It follows that the main category of machine learning algorithm used in activity recognition is that of classification and as such will be discussed and demonstrated in this thesis.

Classification algorithms can be categorized into generative and discriminative models (Jordan, 2002), where the differentiating principles between the two can be defined by the probability distribution over the feature space; in essence the generative approach models the joint probability $p(x,y)$, where x is the feature vector and y is the target label, and the prediction for the most probable label is achieved using Bayes rule to calculate the probability of y given x , $p(y/x)$. It can be said that the generative model learns the underlying probabilistic structure that generated the data and that this is, in essence, a pre-processing stage before Bayes rule is implemented at the prediction stage. Generative models are popular in the field of activity recognition in ambient environments, examples of which are the NB classifier evaluated by (Tapia, et al., 2004), (Stikic, et al., 2008), (Crandall & Cook, 2008) as well as (Szewczyk, et al.,

2009), HMMs implemented in (Sanchez, et al., 2007), (Stikic, et al., 2008), (Crandall & Cook, 2010), (Roy, et al., 2011) and more recently (Kabir, et al., 2016), Gaussian Mixture Models (GMM) proposed in (Austin, et al., 2011) and (Plotz, et al., 2011). In contrast the discriminative model seeks to find a boundary that represents the separation of the labelled classifications in the feature space. This can be thought of as modelling the probability of y given x directly $p(y/x)$ and so a prediction of the label for a yet unclassified input vector is made by evaluating at which region in the segmented feature space the vector lies. The distance between a data point from the decision boundary provides a confidence measure of how sure the model is of the classification provided. Examples of discriminative models are Logistic Regression (LR) evaluated in the field of HAR by (Riboni & Bettini, 2009), (Kwapisz, et al., 2011) and (Al-Bin-Ali, et al., 2003), SVM by (Luvstrek & Kaluza, 2009), (Cook, et al., 2013) and (Nef, et al., 2015), Decision Trees (DT) used in activity recognition by (Logan, et al., 2007), (Riboni & Bettini, 2009) and (Luvstrek & Kaluza, 2009), Random Forests (RF) have been used to classify activities in (Bayat, et al., 2014) and (Nef, et al., 2015), and Adaboost by (Lester, et al., 2005), (Luvstrek & Kaluza, 2009), (Keally, et al., 2011) and (Wen, et al., 2015). The following sections discuss these Machine Learning algorithms in more detail.

3.1.1.1 Naïve Bayes (NB)

A key idea in the field of machine learning is the ability for algorithms to deal with uncertainty. Whenever models are employed to represent real world phenomenon there will always exist some uncertainty in the model, as is quoted in (Box & Draper, 1987):

“Essentially, all models are wrong, but some are useful”

Useful models are designed to deal with uncertainty, whether this comes from noisy sensors, stochastic influences, or from datasets that are too small to represent a population and hence the model is created with incomplete knowledge. One of the most useful theories used in machine learning is that of conditional probability which is based around the premise, to use an example from the field of activity recognition, that the probability of the current activity label being “cooking” can be calculated given the incoming sensor data. This probability provides a level of

certainty as to which activity is taking place, and if the next sensor event indicates that the “cooker” is in use, the original probability can be revised based on this new evidential data. This concept is known as the Bayesian interpretation of probability (Bishop, 2006) and uses Bayes Theorem (Joyce, 2008) to convert a probability calculated using data observed before any new data (prior probability) into a probability conditioned on new observable data (posterior probability). In essence Bayes Theorem computes the posterior probability $P(H|D)$ of a hypothesis (H) given the data (D), by computing the probability of the data given the hypothesis $P(D|H)$, multiplied by the probability of the hypothesis $P(H)$. The result of this calculation does not in fact produce a probability, but a true probability is given by normalising the result with a division by the probability of seeing the data, thus the equation for Bayes theorem takes the form:

$$P(H|D) = \frac{P(H) \cdot P(D|H)}{P(D)} \quad (3.1)$$

The NB classification algorithm is a generative, supervised algorithm that is based on Bayes Theorem, but is a simplified version that uses the naïve assumption that the features used to represent the data are conditionally independent. This is useful because it reduces the number of calculations necessary to compute $P(H|D)$ when we consider that D is a vector of features x_1, \dots, x_n then Bayes Theorem with multiple features becomes:

$$P(H|x_1, \dots, x_n) = \frac{P(H) \cdot P(x_1, \dots, x_n|H)}{P(x_1, \dots, x_n)} \quad (3.2)$$

From the product rule and chain rule of probability theory calculating $P(x_1, \dots, x_n|H)$ results in $O(|D|^n \cdot |H|)$ parameters and so as the number of features and classes of activity increase the computational complexity increases. As the NB classifier learns the conditional probability of each feature given the class from training data, by using the assumption of conditional independence of features given the class, the number of parameters is reduced and so the computational complexity is reduced. The effect of this simplifying assumption is that the

calculation for the probability of the features given the class $P(x_1, \dots, x_n|H)$ is represented by the product of the probability of each feature given the class:

$$P(x_1, \dots, x_n|Y) = P(x_1|Y) \cdot \dots \cdot P(x_n|Y) \quad (3.3)$$

As the denominator in equations (3.1) and (3.2) is effectively a normaliser that ensures the resulting value is a probability, the most probable H can be determined by the maximum of each calculation per class (*argmax*) and so the NB equation becomes (Friedman, et al., 1997):

$$\operatorname{argmax}_Y P(Y|x_1, \dots, x_n) = \max_Y P(Y) \prod_{i=1}^n P(x_i|Y) \quad (3.4)$$

As the features used in the field of HAR are generally numerical such as used in the “bag of sensors” representation shown in (Cook & Krishnan, 2015) . The form of the NB algorithm used in HAR is generally that of the Multinomial NB classifier (Pedregosa, et al., 2011). This version of the NB classifier is parameterised by a Laplace smoothing parameter (alpha) that accounts for features that are not available in the labelled training data and so prevents zero probabilities.

The NB classifier has been used by (Tapia, et al., 2004) where the reported accuracy for the classification of 22 human activities monitored in an ambient environment lie in a range from 25% to 89% depending on the activity, no detailed performance data is available for the results so further evaluation of the NB classifier on the same data is carried out in chapter 4. The NB classifier was also used by (Crandall & Cook, 2008) where classification accuracies over 95% were reported and false positives of 2% claimed to be possible using extra features formed from the hour of the day and day of the week along with careful segmentation of activities. (Stikic, et al., 2008) Used the NB classifier to determine the accuracy of accelerometer data for the classification of IADL's and reported accuracies of approximately 67%. (Szewczyk, et al., 2009) Also used data recorded from the same environment as (Tapia, et al., 2004) but the focus was on how effective different types of annotation of labelled training data were on the accuracy of a classifier, the researchers report a best accuracy of 73.6% using the NB classifier with

visualization and resident feedback as the annotation mechanisms. A publication by (Nef, et al., 2015) describes a study using PIR and light sensors to classify 8 activities using several classifiers and concluded that the NB classifier performed the worst with an average F1 measure of 27.88 compared to the SVM and RF classifiers with average F1 measures of 41.23 and 71.33 respectively.

3.1.1.2 Logistic Regression (LR)

The LR (Cox, 1958) classifier is a discriminative algorithm used to discover a linear separation between classes in the feature space, determined by the training data (D). Logistic Regression effectively models the probability distribution of a class using the coefficients of a feature vector and applies a function that restricts the output of the algorithm to a range of probabilities [0,1]. The sigmoid function shown in equation (3.5) is a common function that is often applied to the linear combination of the feature vector and coefficients to map the real numbered output to within 0 and 1 (Abu-Mostafa, et al., 2012):

$$p(y = 1|x; \omega) = \frac{1}{1 + e^{-\omega^T x}} \quad (3.5)$$

Where y is the classification label, $x \in \mathbb{R}^n$ is the feature vector and $\omega \in \mathbb{R}^n$ are the coefficients or weights that are learned from the training data using the LR algorithm. The algorithm is an optimisation problem based around finding the probability that maximises the likelihood of the training data; this is achieved using the maximum likelihood function shown in equation (3.6):

$$\arg \max_{\omega} \sum_{i=1}^N \log p(y_i|x_i; \omega) \quad (3.6)$$

LR has been predominantly used for physical activity recognition in the field of HAR, for example research conducted by (Riboni & Bettini, 2009) evaluated the use of LR against other classifiers for the recognition of 10 activities using accelerometer and GPS data and report an accuracy of 80.21% against 68.55%, 71.81% and 66.23% for that of NB, SVM and DT (C4.5)

respectively. Work carried out by (Kwapisz, et al., 2011) used LR for the classification of 6 physical activities using accelerometers on mobile phones providing a reported recognition accuracy of 78.1%. An older study, but arguably of more interest in this thesis, was presented by (Al-Bin-Ali, et al., 2003) and used the LR algorithm in an initial effort to prove the usefulness of the algorithm to correlate the sensors used to human activity recognition in ambient environments. The experimentation used a conference hall and the state of association between wireless nodes and access points to determine the activities of the occupants. Although the results were inconclusive the idea is built upon in this thesis in attempt to discover the sensors that contribute most to activities of interest in an ambient environment.

3.1.1.3 Support Vector Machines (SVM)

The SVM classifier (Boser, et al., 1992) is a discriminative classification model that builds upon the principles of LR but uses a hyperplane to separate distinct classes of examples. Where the aim of LR is to maximise the probability of the data, the hyperplane in an SVM is chosen in such a way as to maximise the margin between the two classes in an attempt to minimise the generalisation error. The SVM uses borderline data points known as support vectors to discover the maximum margin once the decision boundary between classes has been established.

The resulting model, which is determined by the selection of the support vectors that in turn determine the chosen hyperplane, is parametrised by weights and used to classify new examples. In the case where the classification of examples is not a clear linear separation, then the SVM can be designed to map the input feature vector into a higher dimensional, non-linear feature space. The separating hyperplane is then implemented within the higher dimension. This technique is known as the kernel trick and enables the testing of various non-linear kernels to determine which kernel produces the clearest separation in the higher dimension; chapter 5 describes some of the most common kernels used during the implementation of a smart water meter.

To classify between many classes a series of one versus all classifications are implemented (Hsu & Lin, 2002), note this is also the strategy used in LR for multi label classification problems. SVMs are essentially a quadratic optimization problem and thus have a high algorithmic

complexity with the memory requirements proportional to the number of training examples, so provide a promising option when the number of training examples is limited as is the case most often with human activity recognition. SVMs have been compared with a range of machine learning algorithms for fall detection and activity recognition by (Luvstrek & Kaluza, 2009), the researchers used coordinates from body tags to collect data during 5 general movement activities one of which was falling and reported that an SVM produced the most accurate classifier with 97.7% classification accuracy, it should be noted that the accuracies of other machine learning algorithms i.e. NB, DT, RF, and Adaboost showed comparable results of 89.5%, 94.1%, 97% and 97.7% respectively and so more detailed machine learning evaluation and performance metrics may be needed to differentiate between the classifiers. The research published by (Cook, et al., 2013) compared the performance of an SVM with that of a NB classifier and reported a classification accuracy of 91.52% against 90.82% respectively. The researches then went on to use the SVM to classify activities, using an online activity recognition procedure, with promising results. The promise of the SVM for activity recognition is backed up by (Nef, et al., 2015) where it was concluded that the SVM classifier performed better than the NB classifier with an average F1 measure of 41.23 compared to that of NB with an F1 measure of 27.88 but faired considerably worse when compared to that of RF classifier which produced an F1 measure of 71.33.

3.1.1.4 Decision Trees (DT)

The DT (Quinlan, 1986) is a discriminative, hierarchical classification model that is represented by a tree structure implementing sequential splits on the data that are based on heuristics of the features. These splits are referred to as decision nodes with each decision node implementing a test function $f_m(x)$ that results in a discrete outcome represented by output branches. Leaf nodes are the termination point and, in the case of classification, represent a label y corresponding to the final classification. Each $f_m(x)$ at the nodes represents a function that discriminates between the input data points in order to subdivide the input space (Alpaydin, 2014).

Discovering the optimal split for a DT is NP-Complete (Hyafil & Rivest, 1976) and so to produce a local optimal split, heuristic are implemented. Greedy top down algorithms are the most frequent techniques used in the literature with the most popular being the Iterative Dichotomiser

3 (ID3) developed by (Quinlan, 1986) specifically for use with categorical features, the successor to ID3 referred to as C4.5 (Quinlan, 1993) which included specific improvements to handle continuous features, and the Classification and Regression Tree (CART) algorithm developed by (Breiman, 1984) as both a classifier and a regression algorithm that is able to generate regression trees that predict real values or categorical labels.

The univariate statistical test function used to determine how well the feature at each node classifies the data are determined by the algorithm; the ID3 algorithm uses information gain (Mitchell, 1997) as the statistical test and stops when all data in a node belong to a single class or the best information gain is zero, the C4.5 algorithm uses gain ratio (Witten, et al., 2011) as the statistical test and stops when a data point count reaches a threshold, the CART algorithm (Witten, et al., 2011) also uses information gain and the tree is grown fully and then optionally pruned by cost-complexity pruning to reduce the complexity of the model. In the research by (Riboni & Bettini, 2009) and (Luvstrek & Kaluza, 2009) the DT performed better than NB but are generally out performed by SVM and LR classifiers in both cases. In a research paper by (Logan, et al., 2007) the researchers found that the DT classifier performed well when used to classify activity data from infrared motion sensors although the machine learning performance metrics are not reported in detail.

3.1.1.5 Random Forest (RF)

The RF Classifier (Ho, 1995) implements the idea of Bootstrap Aggregation, often referred to as Bagging (**B**ootstrap **A**ggregating) and was first proposed by (Breiman, 1996). The idea proposed to use an ensemble of base-learners rather than a solitary learner with classifications made by aggregating the output of the ensemble. Bagging is an aggregation method by which base-learners in the ensemble are trained on training data sampled randomly, with replacement, from the original training data, thus generating a different dataset of the same size as the original for each learner. Random sampling with replacement results in randomly chosen duplicate data points, which in turn means that each bootstrap sample will have variations in those data points that are duplicated and as the datasets are the same length as the original, then it follows that, each randomly chosen dataset will have different missing data points and so each dataset is subtly

different. DT algorithms are a common base-learner used in bagging as they are very sensitive to small changes in the dataset, which can potentially result in a different sequence of features being chosen to make the optimal splits. The RF classifier combines bagging with DT classifiers, but also implements random selection of a subset of features per base-learner often called subspace sampling or feature bagging proposed by (Ho, 1998). RF classifiers were compared with NB, SVM and Adaboost classifiers by (Luvstrek & Kaluza, 2009) where they performed comparably to an SVM and outperformed both NB and Adaboost. They have been implemented in activity recognition using body worn sensors by (Bayat, et al., 2014) where data from an accelerometer in a mobile phone was used to classify six movement activities and the results compared with LR and SVM classifiers. The study reported accuracies of 87.55% for RF, 85.41 for LR and 88.76 for the SVM classifier. The study of activity recognition in ambient environments undertaken by (Nef, et al., 2015) shows extremely promising results when used with simple binary sensors where the RF classifier outperformed the SVM classifier and the NB classifier with an average F1 measure of 71.33 compared to 41.23 and 27.88 respectively.

3.1.1.6 Adaboost

The Adaboost (**Adaptive Boosting**) classifier introduced by (Freund & Schapire, 1995) is based on the idea of Hypothesis Boosting and implements a number of DT classifiers in an attempt to convert an ensemble of weak classifiers into a strong classifier. The source of the idea can be attributed to a question posed by (Kearns, 1988) which asked:

“Can a set of weak learners be combined to create a stronger learner?”

The question was affirmed in a publication by (Schapire, 1990). The principle behind adaptive boosting is to use an ensemble of sub-learners where each learner added to the ensemble is biased to pay more attention to those data points that were misclassified by the previous sub-learner. The mechanism used to perform the bias on the data points uses weights for each data point in the dataset $\omega_i \geq 0$ initialised to $\frac{1}{N}$. The algorithm works by iterating over the dataset and adding sub-learners until a number, predefined by the user, have been added. Each sub-learner is used to classify the data points in the dataset and calculate the total classification error. The error ϵ is

calculated by summing over the weights ω_i of the data points where the classification of the data point is incorrect. The algorithm then increases the weights of the misclassified data points by using equation (3.7) and decreases the weights of the correctly classified data points using equation (3.8):

$$\omega_i \leftarrow \omega_i \times \left(\frac{1}{2 \times \epsilon} \right) \quad (3.7)$$

$$\omega_i \leftarrow \omega_i \times \left(\frac{1}{2 \times (1 - \epsilon)} \right) \quad (3.8)$$

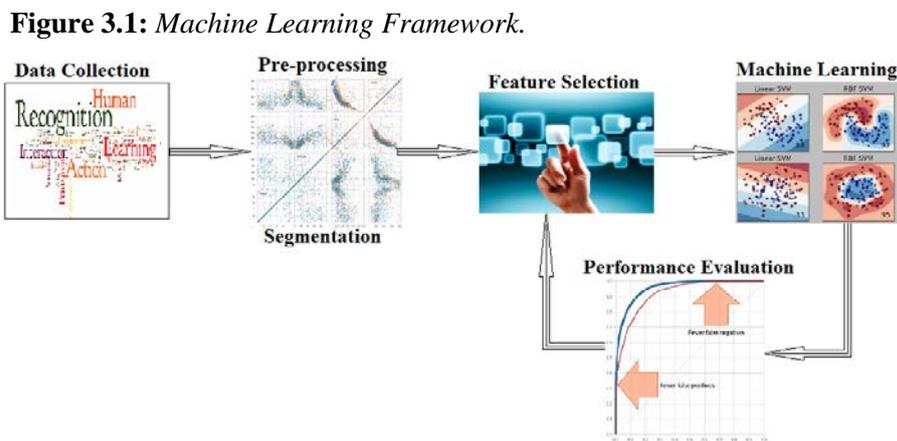
This iterative adaptation of the data over the sub-learners ensures that weights on the data are increased for miss-classified data points and decreased for correctly classified data points. This in turn ensures that subsequent sub-learners focus more on miss-classifications and less on data points that have already been classified correctly. Finally the algorithm calculates a confidence factor α for each sub-learner in the ensemble:

$$\alpha \leftarrow \frac{1}{2} \times \log_e \left(\frac{1 - \epsilon}{\epsilon} \right) \quad (3.9)$$

When the learning phase is complete predictions are made using a weighted aggregate implementing the confidence factors associated with each sub-learner (Kelleher, et al., 2015). An interesting application of Adaboost by (Lester, et al., 2005) reports its use as a method for feature selection to improve classification by a downstream generative algorithm. Adaboost has also been implemented in fall detection and activity recognition by (Luvstrek & Kaluza, 2009) with reported results that are marginally worse than an SVM and a RF but outperform DT and NB classifiers. The work by (Keally, et al., 2011) also demonstrates the use of Adaboost in feature selection using body worn sensors concluding an improvement in accuracy using a downstream generative algorithm after Adaboost feature selection. The research undertaken by (Wen, et al., 2015), also using body worn sensors and Adaboost for feature selection reports that learning is improved by a reduced feature set adapted from an Adaboost classifier. This thesis builds on these ideas and also claims a contribution for the use of Adaboost as both a feature selection technique and classifier for activity recognition using binary sensors in an ambient environment.

3.2 Machine learning analysis framework

Machine learning systems of classification using supervised learning algorithms generally follow a predefined framework used to develop the machine learning model, off-line, that will ultimately be used for on-line classification in the field. This off-line machine learning framework closely matches frameworks used for most fields of machine learning when the aim is to analyse specific details of a machine learning project. This differs from the on-line process referred to as the Activity Recognition Chain described in (Bulling, et al., 2014) where the aim is to create a processing pipeline after analysis has taken place. Figure 3.1 depicts the elements of a suggested framework that is convenient for the discussion of the feature selection process that is the focus of the next chapter.



The framework is initialised by collecting the labelled data that is relevant to the activities of focus, in the case of HAR in ambient environments this is the raw data from sensors within the environment and is collected over a time duration that is often optimised to maximise the quantity and balance (Japkowicz & others, 2000) of activities. This raw sensor data is often not in a suitable format for direct use in a classification algorithm and so a pre-processing stage is used to transform the data (Cook & Krishnan, 2015), this may involve dealing with missing or corrupt data (Witten, et al., 2011) or a complete statistical transformation of the features x may be applied that attempts to improve the classification process (Cook & Krishnan, 2015). When using an off-line analysis framework to investigate the possible machine learning models that best approximate the target function f , as well as determine the features from X that provide the most

appropriate basis for classification of the activities in the target Y , it is important that consideration is given to the technique used for segmentation of the on-line feature vector \mathbf{x} , when used as the input vector to the machine learning algorithm for classification of unseen input data. Segmentation is a pre-processing technique that is used to determine the boundaries of the data, in the hope that a vector \mathbf{x} is produced that includes the relevant features that describe a label y , and is often included in the pre-processing stage of the analysis framework.

Often the sensor data that are collected, or the features produced during pre-processing and segmentation are not the optimum choice to determine the most accurate mapping to an activity label, this is effectively the problem of choosing the input space (X), in this case feature selection can be performed to determine those features that contribute most to the selection of g and provide the closest approximation of f . If the data has a large number of features then feature selection can be performed to reduce the dimensionality of \mathbf{x} and hence the computational complexity of computing the function g . More importantly the dimensionality of the data is exponentially related to the number of training data points required to achieve a reasonable approximation of f (Friedman, 1997). As one of the challenges of HAR in ambient environments is the limited quantity of labelled training data, as highlighted in chapter 2, it follows that by reducing the dimensionality of the input feature vector, the quantity of labelled training data necessary to approximate f may be reduced (Ng, 2004). Once the function g has been modelled it can be used to determine the most probable label for new, previously unseen, feature vectors. In order to evaluate the resulting model and to determine, through comparison, the most appropriate combination of features and learning algorithm, the final stage of the framework uses performance metrics to evaluate and compare these choices for selection in the final on-line activity recognition system.

3.2.1 Data collection

In general the target labels for machine learning applications are determined by the context of the problem at hand and because the learned models represent a function g , that maps the feature vector \mathbf{x} to the target label y , the data that is collected will determine the utility of the models

created. Within the field of HAR in ambient environments the raw data is collected from sensors affixed to the environment and often the decisions, that will determine which sensors to use, are made based on the criteria of availability and suitability of sensor types. There are limitations on the types of sensors that can be usefully placed within an environment, but ultimately the performance of the learned model will depend on the selection of these sensors (Tapia, et al., 2004). Analyses of the models and features using a feature selection stage within the analysis framework, may help in the decision making process by indicating the relevance of each sensor to the accurate classification of the chosen activities.

The features produced from the pre-processing stage of the framework are used to form labelled training and test sets which are in turn used to train and evaluate the machine learning models. The training set is used to train each model, using varying combinations of model parameters and features and in this case is often referred to as a combination of a training and validation set. The test set is a portion of data that is held out and used to evaluate the final chosen model in an attempt at gauging how well it classifies out of sample data points. Both validation and the final evaluation on the test set need suitable machine learning evaluation and performance metrics to aide comparison and ultimately to choose the most suitable learning algorithms for classification of activities. It follows then that the data collected for off-line training and evaluation of the model is annotated with the target activity labels and termed labelled training, validation and test data. The form of the data collected for discussion in this thesis is defined in chapter 2 and relates directly to the field of HAR and related work discussed in the introduction.

Table 3.1 *Sample of data from (Tapia, et al., 2004)*

	Activity label	date	Activity Start	Activity End		
Activity	Toileting	04/03/2003	17:30:36	17:46:41		
Sensor ID	100	68				
event Name	Toilet Flush	Sink faucet-hot				
Event start time	17:39:37	17:39:46				
Event end time	18:10:57	17:39:52				
Activity	Preparing Lunch	04/01/2003	11:21:17	11:38:22		
Sensor ID	140	137	131	53	84	131
event Name	Door	Freezer	Toaster	Cabinet	Drawer	Toaster
Event start time	11:23:04	11:23:55	11:24:08	11:34:59	11:35:04	11:35:12
Event end time	11:23:07	11:24:03	11:24:14	11:35:01	11:35:07	11:35:22

In order to provide clarity to the discussion and further relevance to the experimentation that follows, Table 3.1 shows a sample of raw labelled data taken from the work carried out in (Tapia, et al., 2004) where binary sensors were affixed to everyday objects and triggered on opening-closing and activation-deactivation events. The data is analysed in chapter 4 to provide some evidence as to those sensors and hence the fixtures and fittings that prove most relevant to the classification of activities in ambient environments.

This sample data highlights the need for pre-processing to organise the data in a manner more convenient for representation in the vector form suitable for use by machine learning algorithms. Sample data taken from a study carried out by the CASAS group (Cook & Schmitter-Edgecombe, 2009) is shown in Table 3.2. This data is a sample taken from a dataset that represent the sensor events recorded as occupants of an ambient environment perform five scripted activities chosen from the IADL scale. The data is captured from a combination of motion sensors, item sensors (medicine containers, spoons), reed sensors (placed on cabinet doors), water sensors (to determine the water flow through taps), burner sensors (to detect gas usage) and the asterisk open source telephone exchange to detect telephone usage. For comparison this data will also be used to help provide some evidence to indicate those sensors that prove most relevant to the classification of activities.

Table 3.2 *Sample of data from (Cook & Schmitter-Edgecombe, 2009)*

Date	Time	Sensor ID	Data	Activity label
2008-02-27	12:49:52.624433	M14	ON	Wash_hands begin
2008-02-27	12:49:53.802099	M15	ON	
2008-02-27	12:49:54.24004	M16	ON	
2008-02-27	12:49:55.470956	M17	ON	
2008-02-27	12:49:55.470956	M15	OFF	
2008-02-27	12:49:55.808938	M14	OFF	
2008-02-27	12:49:57.548709	M16	OFF	
2008-02-27	12:49:57.717712	M13	OFF	
2008-02-27	12:49:58.10558	AD1-B	0.0332818	
2008-02-27	12:49:59.197328	M17	OFF	
2008-02-27	12:50:01.10764	AD1-B	0.302934	
2008-02-27	12:50:10.25482	AD1-B	0.471413	
2008-02-27	12:50:25.21543	AD1-B	0.538329	
2008-02-27	12:50:38.708635	M17	ON	
2008-02-27	12:50:40.5204	AD1-B	0.467429	
2008-02-27	12:50:42.521531	M17	OFF	Wash_hands end

3.2.2 Data pre-processing and Segmentation

The pre-processing stage of the framework takes in the raw sensor data and processes this data into a form that is more suitable, as both a representation that best describes the activity and as data that can be handled by a machine learning algorithm. There are numerous representations of features that describe the activities in the field of HAR both in terms of the sensors used and the pre-processed features (Cook & Krishnan, 2015). A discussion of these feature types and ultimately the choice of feature that best represents a sensor for feature selection will follow but prior to this discussion it is necessary to provide a brief overview on the concept of segmentation.

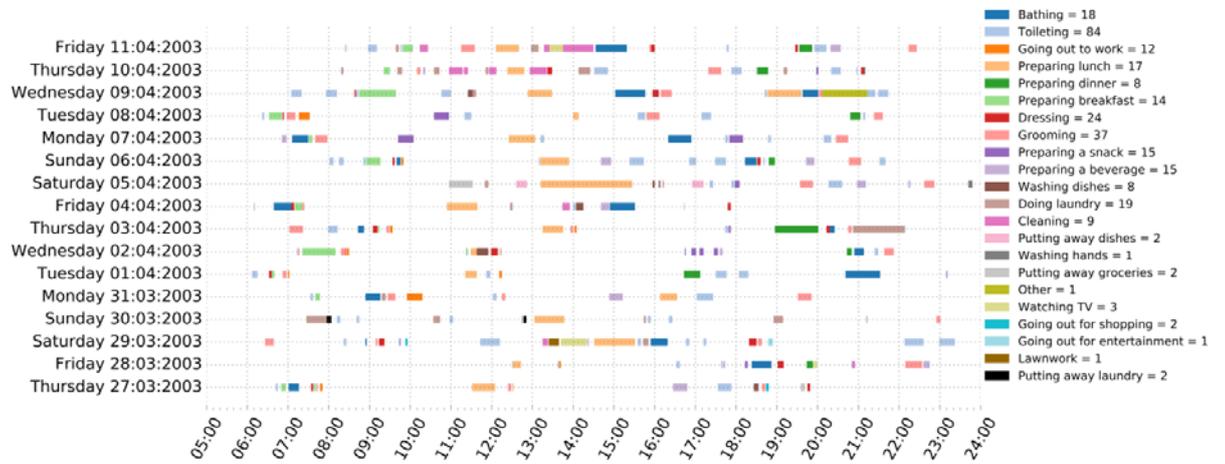
In order for an on-line machine learning system to effectively classify a particular feature vector representation of the sensor data, the system must include a mechanism for selecting those sensor events from an incoming stream of sensor data. This process is known as segmentation (Cook & Krishnan, 2015) and aims to select the sensor data that lies between activity boundaries. In systems where the sensors produce high definition data, such as when accelerometers are used for activity recognition, as described in (Fuentes, et al., 2012) and (Bayat, et al., 2014) it is common to use a fixed window of time to segment the data. With a fixed window and high resolution data, there is a high probability that enough sensor data will be collected to capture an activity (Krishnan & Cook, 2014). This type of fixed window segmentation is often challenging to implement in ambient environments because the events are triggered by human activity rather than a constant sampling rate and can often be very sparse in distribution over time (Krishnan & Cook, 2014). Some studies have used overlapping windows that take the values of sensors observed in previous windows if no subsequent value is recorded, thus reducing the sparsity of the feature vector (Logan, et al., 2007). An alternative segmentation method known as sensor based segmentation uses windows sized on the number of sensor events rather than a time period (Krishnan & Cook, 2014) and has proven to be suitable for use in ambient environments. Segmentation is an essential part of the on-line machine learning process but, as the scope of this work is to investigate and identify those sensors that contribute most to the classification of activities. An off-line approach that uses pre-segmented data is more suitable and is a common approach in the field when the focus of the work is on the upstream elements of the machine

learning framework. The data shown in Table 3.1 and Table 3.2 has been collected and annotated by a human annotator and so the segmentation process has been carried out manually off-line, this process is also used in the final chapter.

Often in the field of HAR it is possible to use the raw data values from sensors when they are appropriate for use with the chosen classification algorithm, for example in (Wilson & Atkeson, 2005) Bayes filters and particle filters are used with data that is represented as the binary value of simple sensors during a segmented time window. On the other hand, sometimes it is impractical to use raw sensor data for the input to a machine learning algorithm and so some thought must be put into the manipulation or pre-processing of this data before classification can be achieved. As discussed, the data used in this thesis is pre-segmented and annotated and as such can be pre-processed using a data type that relates directly to an activity, this is often not the case in on-line systems where pre-processing may need to be carried out using the data regardless of whether a particular feature relates to an activity. As such this segmented data is often manipulated into a form that can be represented as a vector of features that best describe the activity; this manipulation may involve dealing with missing or incomplete data or generating new features that allow more flexibility in modelling the function g . Often the type of pre-processing required is governed by the data at hand, and assumptions may need to be made about this data from prior analysis. Using the data shown in Table 3.1 as an example; this data includes recorded sensor events along with the start and end time of each event, triggered during a specific activity, but does not indicate the form of the data from that event. Although documentation provided with the data confirms that the sensor types used, were reed switch sensors and the data type produced from such sensors is Boolean 'ON', 'OFF' [1,0], therefore assumptions regarding the sensor data need to be made in this case. It follows that analysis of the data is essential to discover any issues there may be with the data as well as underlying patterns between features and activity labels that may help in the engineering of better features. For example Figure 3.2 shows the (Tapia, et al., 2004) dataset using a plot of the activities as they occurred against date and time, the key to the right indicates the activity label and the number of instances of each label within the dataset. If machine learning models are to be trained and tested using this dataset, it is clear that activities

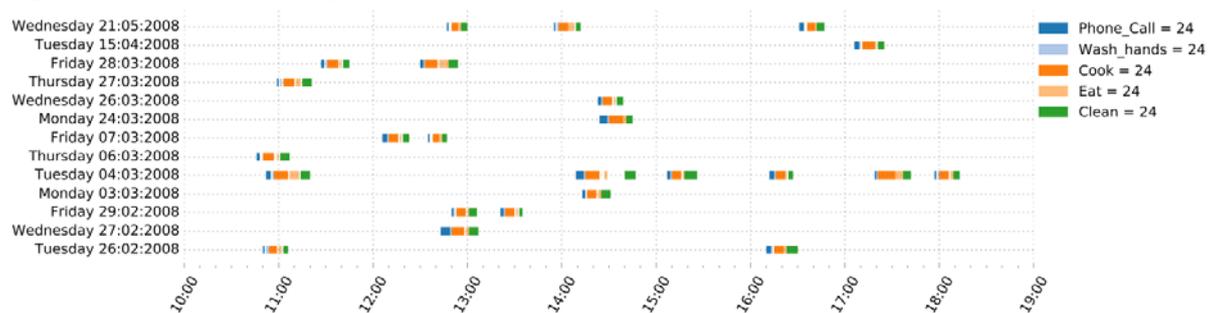
with very few instances may not be suitable candidates for classification in isolation and may need to be omitted from the modelling process or grouped into the “Other” activity label.

Figure 3.2 Daily activities generated from the Tapia Data set.



In contrast the plot of activities over time and date of the (Cook & Schmitter-Edgecombe, 2009) dataset, shown in Figure 3.3 represents very clean, well balanced data that is ideal for use in training and testing machine learning models. It should be noted however, that this dataset represents a relatively small sample and may not be representative of the entire sample set. This may result in models produced that do not perform well on out of sample data generated in a less clinical setting, and it is highly likely that the models produced will over fit the training data.

Figure 3.3 Daily activities generated from the CASAS Dataset.



In the field of HAR for ambient environments there are a wide range of possible features that can be engineered from the raw sequence of sensor data; these can be based on the activity temporal cycles as demonstrated in (Crandall & Cook, 2008) and (Szewczyk, et al., 2009), state changes as demonstrated in (Kabir, et al., 2016) and (Van Kasteren, et al., 2008), activity sequence statistics also used in (Fang, et al., 2014) and (Van Kasteren, et al., 2008), the type and number of

events in each activity sequence often referred to as “bag of words” and shown in (Yamada, et al., 2007) or more appropriately referred to as “bag of sensors” in (Cook & Krishnan, 2015), or the context of the sequence in terms of those activities prior to and after the current activity also demonstrated in (Fang, et al., 2014). For example (Crandall & Cook, 2008) found that by generating a feature that represents the hourly temporal cycle and combining this with sensor state values, the overall accuracy for classifying events increased and false positives were reduced, when compared to using no temporal data. A study undertaken by (Szewczyk, et al., 2009) used the “ON”, “OFF” state of each sensor along with discretised event timings binned at intervals of “morning”, “afternoon”, “evening”, and night as well as features based on the hour of the day the activity took place.

The focus of this chapter is to provide background for subsequent analysis to determine the contribution of each of the sensors in the (Cook & Schmitter-Edgecombe, 2009) and (Tapia, et al., 2004) datasets. In order to make an unbiased analysis, the raw sensor data is minimally processed to produce simple sensor counts based on the ‘ON’ trigger of each sensor. This method is often used in natural language processing (Salton & McGill, 1983) and is referred to as “bag of words” in an attempt to indicate that the sequence information associated with the events is overlooked. Table 3.3 shows the sensor count representation of the activities sample taken from the (Tapia, et al., 2004) dataset, note that the assumption was made that the data is binary and the start time of each sensor event represents the ‘ON’ state and it is this state that is counted here.

Table 3.3 *Sensor count representation of sample data taken from unbalanced dataset*

Date	Start_time	End_time	100	68	140	137	131	53	84	Activity
04/01/2003	11:21:17	11:38:22	0	0	1	1	2	1	1	Preparing Lunch
04/03/2003	17:30:36	17:46:41	1	1	0	0	0	0	0	Toileting

Table 3.4 shows the sensor count representation of the activities sample taken from the (Cook & Schmitter-Edgecombe, 2009) dataset, note here that the continuous data representation for the water flow sensor (AD1-B) has been converted into a feature that represents the ‘ON’ state effectively creating a sensor event when human activity has triggered the initial flow of water. This decision was taken as the hypothesis of this thesis includes the locality of the resident in the

environments and it is possible that the occupant can turn on a tap, leave the room for a time, and return to turn off the tap, therefore it is argued that the on and off events only, provide an indication of locality.

Table 3.4 *Sensor count representation of sample data from balanced dataset*

Date	Start_time	End_time	M13	M14	M15	M16	M17	AD1-B	Activity
2008-02-27	12:49:52	12:50:42	1	2	2	2	4	5	Wash Hands

3.3 Machine Learning Evaluation and Performance Metrics

Evaluating the performance of machine learning algorithms is an essential process that is used both to determine the optimum parameters for a particular classification algorithm as well as the most suitable algorithm to solve a particular classification problem. The context of the activity recognition problem is therefore tightly linked to the choice of the evaluation metric and the stochastic nature of human activity poses problems which become apparent in practical deployments of HAR systems (Minnen, et al., 2006), (Ward, et al., 2011). The most common performance metric used in the literature for HAR systems is that of classification accuracy, i.e. the fraction of correctly classified instances given by the sum of True Positives (TP) and True Negatives (TN), divided by the total population of data points (N):

$$\text{Classification Accuracy} = \frac{TP + TN}{N} \quad (3.10)$$

The problem with using classification accuracy in the field of HAR is that the datasets collected from ambient environments in natural settings are inherently unbalanced (Tapia, et al., 2004), it is common, for example, to find datasets with many more “Toileting” activities than “Ironing” activities. If a dataset was made up of 80% “Toileting” activities and a classifier was trained to memorise the “Toileting” activity perfectly it would produce 80% classification accuracy without consideration or classification of any other activity in the dataset. Subsequently in this thesis, the following additional performance metrics are used to enable a more detailed analysis and comparison of classification models:

Precision (Powers, 2011) is a measure of the correct event classifications (True Positives) of a classifier divided by the total event classifications whether correct or not, given by the sum of True Positives and False Positives:

$$\text{Classifier Precision} = \frac{TP}{TP + FP} \quad (3.11)$$

Therefore precision is a measure of the fraction of positively classified data points that were correct, in other words, precision is a measure of how precise the classifier is at identifying a class label against all other classes in the data set. Note: the higher the count of false positives the lower the precision.

Recall (Powers, 2011) is a measure of the correct event classifications (True Positives) divided by the sum of True Positives and the data points that were classified with another label but should have been classified positive i.e. False Negatives (FN). That is those labels in the class were missed by the classifier:

$$\text{Classifier Recall} = \frac{TP}{TP + FN} \quad (3.12)$$

The Higher the FN count the lower the number of class labels can be detected and the lower the recall of the classifier. Note that these measures only focus on the positive data points but this is relevant when the classifiers used apply a one vs many process.

The F1 Measure (Powers, 2011), often referred to as the F-measure, is a balanced composite measure of both precision and recall that enables a single number evaluation of a classifier and is defined by.

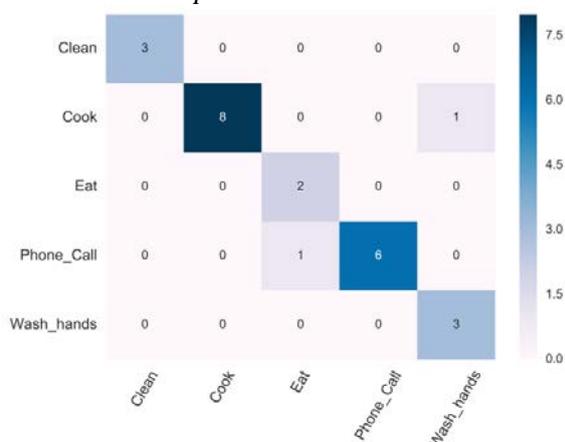
$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3.13)$$

These metrics provide a general overview for comparison, of the classification performance for models over each of the class labels and when combined with the F1 measure provide an overall

impression of the best performing model across the range of class labels. These metrics do have limitations when trying to determine exactly which events are causing confusion in the classification process, therefore confusion matrices are used to get an overall impression of the confusion (i.e. misclassification) in an HAR system.

A confusion matrix is designed to show the correct and incorrect classification of labels by the classifier, an example of which is shown in Figure 3.3 illustrating 5 activities. The diagonal cells shown from top left to bottom right indicate the correct classification of an activity. The rows indicate the true classification (ground truth) and the columns indicate the classification estimated by the model. It can be seen from the matrix that 2 activities out of a total of 24 were misclassified, a “Cook” activity was classified as “Washing Hands” and a “Phone Call” activity as an “Eat” activity, such a graphic provides some detailed information on where the classifier is confused and so is useful in this thesis to determine the link between specific sensors and effective classification.

Figure 3.3 Confusion Matrix produced from balanced sample data.



ROC curves or receiver-operator-characteristics are a common tool to illustrate the impact of parameters on classification approaches. They can be implemented when considering the importance of certain outputs from the system. For example, a system aimed at monitoring elderly people in a single occupancy environment may place a high importance on the classification of a “Leaving Home” event. In this scenario it may be a requirement that the machine learning model is sensitive to this activity, at the cost of false predictions in other less

important activities. When evaluating results for such a model it is often more appropriate to use a performance metric that highlights the overall difficulty of classifying the important events of interest. ROC curves are mentioned here for completion but are not deemed necessary at this stage of the research but will be considered as further research determines those activities deemed critical to the well-being of the occupants of an ambient environment.

3.4 Summary of machine learning in HAR

This chapter has discussed the privacy challenge faced for Human Activity Recognition Systems indicating that activity recognition using binary sensors may be a more acceptable and practical method for use in assisted living applications as appose to the more intrusive vision or body worn sensor techniques. The concept of Machine Learning has been introduced as the dominant technology in the field, with supervised learning and classification being highlighted as the dominant techniques. Examples of how these algorithms have been used and their performance for activity recognition to date have been provided. It has been stressed that for this thesis Machine Learning forms one element of a Machine Learning analysis framework of which it is preceded by data collection, data pre-processing and feature selection. Data collection has been omitted at this stage and will be picked up in Chapter 6 but two publicly available datasets are introduced where simple sensor data was collected from ambient environments and these will be used for analysis in subsequent chapters. Data pre-processing is discussed with a view to justifying the use of simple features that relate transparently to the sensors of interest according to the aims of the thesis, as provided in chapter 1. A detailed discussion of feature selection is deferred to Chapter 4 where analysis of the correlation between sensors and specific activities is carried out. The datasets presented in this chapter highlight the challenges in the field relating to the quantity of labelled training data and the unbalanced nature of human activity data. In light of this it is pointed out that Machine Learning algorithms can bias toward the majority class and so more enlightening metrics, beyond accuracy, are introduced so that more realistic comparisons of algorithm performance can be made.

Chapter 4 Feature Selection for HAR

Chapter 3 provided some background information on an analysis framework that will be used to recognise activities from sensors situated within an ambient environment. The concept of machine learning formed the core of this framework and common methods of data collection, pre-processing and segmentation were described. Datasets from (Cook & Schmitter-Edgecombe, 2009) and (Tapia, et al., 2004) were used to provide context to the discussion and highlight common issues with raw data that may need to be addressed before the data is used for classification of activities. The chapter was used to form a foundation for this chapter which will describe the concept of feature selection and the procedures used to empirically evaluate the relevance of the sensors used to generate the datasets introduced in chapter 3. This is done with a view to gaining insight into the importance of those sensors and ultimately to determine whether motion sensors, electricity usage and water usage may form a sensor subset for activity recognition. This will provide some confidence in the thesis hypothesis that activity recognition may be possible with a reduced number of practically implementable sensors, based on equipment already available in standard environments and so eliminating the need to create fully ubiquitous ambient environments for assisted living.

The concept of feature selection will be defined and described breaking down the commonly used methods into Filter (Liu, et al., 1996), Wrapper (Kohavi & John, 1997) and embedded (Neumann, et al., 2005) methods. In chapter 3 it was pointed out that the (Cook & Schmitter-Edgecombe, 2009) dataset was representative of a balanced dataset that was generated from scripted activities with each activity equally represented with 24 activity data points. Such a dataset is ideal for machine learning algorithms often producing high accuracy measures on both training and test sets as no one activity is able to bias the final model, however such a model may suffer from a bias toward the clean dataset and may not generalise to real world data that is usually more stochastic in nature. Real world HAR data from ambient environments is inherently unbalanced, for example the (Tapia, et al., 2004) dataset where the activity of “Watching TV” occurred only three times and the “Toileting” activity occurred 84 times poses much more of a challenge to

machine learning algorithms that tend to form a bias toward the majority class. This issue is highlighted in this chapter where machine learning classifiers are shown to perform extremely well on the scripted, well balanced dataset and not so well on the more typical unbalanced dataset.

4.1 Feature Selection

In many practical applications of machine learning algorithms, the data that has been pre-processed can result in a large number of features and it is often desirable to reduce this number to enable effective classification beyond the training stage. It is well known that as the number of features represented in the data rises, then the number of data points per class needed for effective training of the model also rises, this is referred to as the “curse of dimensionality” (Friedman, 1997). Furthermore, supervised learning models that have been trained on data incorporating many features, may suffer from overfitting the model to the training data unless some form of regularization (Ng, 2004) and/or feature reduction is implemented. Discriminative algorithms that do not employ regularisation or feature reduction require large training sets, relative to the number of features, to enable the model to generalise well beyond the training data. It follows that by selecting a subset of the features, that contribute most to the correct classification of the activity of interest, the number of training examples required to approximate the unknown target function can be reduced. This is of particular importance in the field of HAR as it may help in addressing two of the main challenges in the field; that of insufficient labelled data and privacy, both of which were discussed in chapter 2. Here we use the feature selection process to address the problem of sensor selection with a view to validating the hypothesis that certain sensors may be more relevant or useful in the recognition of certain activities and can thus be adapted and implemented in a more practical system of activity recognition in ambient environments.

There is some ambiguity, in the field of machine learning, as to the definition of the term important (relevant or useful) with respect to feature selection and what constitutes an important feature. A succinct discussion of which is provided by (Blum & Langley, 1997), where definition

5 which is quoted below, is preferred in this thesis as the aim is to discover a sensor subset that produces comparable classification accuracy to that of the whole sensor set:

“Definition 5 (Incremental usefulness) Given a sample of data S , a learning algorithm L , and a feature set A , feature x_i is incrementally useful to L with respect to A if the accuracy of the hypothesis that L produces using the feature set $\{x_i\} \cup A$ is better than the accuracy achieved using the feature set A ”

And so this definition implies that features have a degree of importance in terms of the comparative accuracy achieved by the model using a feature when compared to not using that feature. This is often referred to as “subset selection” (Alpaydin, 2014) and as stated above, the features used here will be a sensor count representation of those sensors that were activated during an activity and so directly represent the contribution of that sensor.

In summary the aim of feature selection, for the purposes outlined here, is to find a subset k of features from the full feature set d that provides the optimum differentiation between the classes of interest (Alpaydin, 2014). This is in contrast to feature extraction where the aim is to find a new set of k features that are combinations of the original feature set d . Feature extraction is often used to reduce the dimensionality of a feature set by projecting a high dimensional vector onto a lower dimensional space (Guyon, et al., 2006), but because the lower dimension is a transform of the original feature set the transparency of the original feature space is lost and so it is not suitable as a method to determine the contribution of the sensors in an HAR system. Feature selection techniques can be categorised into supervised and unsupervised depending on the use of labelled or unlabelled training data during the selection process. Supervised feature selection is used with classification problems that require subset selection to maximise class discrimination (Li, et al., 2016), therefore this is the feature selection approach used here. Supervised feature selection can be further categorised into Filter Methods, Wrapper methods, and Embedded methods (Chandrashekar & Sahin, 2014).

4.1.1 Filter Methods

Filter methods of feature selection are independent of the machine learning algorithm and are used to assess the statistical characteristics of the features to evaluate their relevance with respect to the classification target and/or their independence from other features. Features that are dependent on other features are often redundant in the presence of those dependent features and those that provide no discriminatory information between target classes are deemed irrelevant or unimportant (Li, et al., 2016). Filter methods aim to score only one redundant feature from the subset of dependent features and zero score irrelevant features. It follows that filter methods of feature selection apply heuristics to evaluate how relevant a particular feature is to the classification of the target as well as a measure of correlation between features. Many filter methods use information theoretic heuristics such as information gain, also methods such as mutual information between each feature and the target are common e.g. for nominal features such as sensor counts probabilities can be calculated from the frequency counts on the data:

$$I(i) = \sum_{x_i} \sum_y P(X = x_i, Y = y) \log \frac{P(X = x_i, Y = y)}{P(X = x_i)P(Y = y)} \quad (4.14)$$

Where $P(X = x_i, Y = y)$ is the joint probability density and $I(i)$ is the measure of dependency between x_i and the target class y . Filter methods tend to be a computationally expensive process when the number of features and the available data is large and are not tuned to any classification algorithm so significant downstream effort may still be required to determine the most appropriate filter/algorithm combination for a specific problem. Filter methods have been applied to activity recognition by (Chen, et al., 2009) Where mutual information was used to evaluate the relationship between each feature, activity pair and those with the highest score were selected for use in a Dynamic Bayesian Network (DBN), the research does not indicate the threshold used to determine the cut off for those features selected and those that were not, this illustrated the need for a downstream process to further determine a cut off threshold as well as model selection. Research by (Fang, et al., 2014) used inter-class distance to determine a features impact on the

classification of an activity. In this case the subset cut off threshold was determined by repeated evaluation of accuracy of subsets trained using an ANN.

4.1.2 Wrapper Methods

Wrapper methods of feature selection (Guyon, et al., 2006) perform an exhaustive search over the subset feature space and evaluate the performance of a specific classification algorithm on each of the subsets. The feature subset that produces the highest performance is then used with the same classification algorithm that was used in subset selection to produce the final classifier. Often an exhaustive search is prohibitive with large datasets that incorporate many features as the search space for N features has complexity of $O(2^N)$ and so NP-hard and computationally intractable (Kohavi & John, 1997) with a high feature count. For example the unbalanced (Tapia, et al., 2004) dataset used here has 72 sensors, if each sensor is considered a feature an exhaustive subset search would require $4.72e^{21}$ separate classifier evaluations (not including cross-validation for parameter selection). For this reason wrapper methods often incorporate some form of optimised search to reduce the complexity of the process, such as best first search or hill climbing both of which were evaluated by (Kohavi & Sommerfield, 1995). Wrapper methods can use forward or backward selection during selection process, the former starting from the empty subset and adding features the latter from the full set of features and subtracting. The forward selection approach is computationally less expensive as classifiers can be evaluated quicker with smaller feature sets however backward selection may capture feature interaction earlier (Kohavi & John, 1997).

Wrapper methods are a viable option when the dataset is relatively small and features are relatively few such as with Human activity recognition in ambient environments, however it is still necessary to develop a search heuristic and suitable evaluation methodology.

4.1.3 Embedded Methods

Embedded methods of feature selection are embedded into the learning process of an algorithm and often play a part in ensuring that the models produced do not simply memorise the training data but can also generalise well to out of sample data i.e. the bias/variance trade off (Friedman,

1997). A good example of this is Regularised Logistic Regression (Ng, 2004) where regularisation is used to tune the LR algorithm; this is achieved by adding a regularisation function $R(\omega)$ to equation (3.6) and so producing equation (4.15).

$$\arg \max_{\omega} \sum_{i=1}^N \log p(y_i|x_i; \omega) - \lambda R(\omega) \quad (4.15)$$

Here $R(\omega)$ is the regularisation function of the model coefficients that penalises large coefficients ω that are indicative of complex models that memorise the training data. This memorisation of the training data is known as overfitting and is described by (Mitchell, 1997) in terms of a hypothesis overfitting the training data if some other hypothesis fits the training data less but performs better on out of sample data. Therefore the regularisation function is used to mitigate overfitting of the model to the training data and the parameter λ can be used to tune the model complexity. When the model is complex it is said to have high variance which refers to the condition when the model fits the particular sample of training data very well. The model representing function g is said to be sensitive to the training sample (Friedman, et al., 1997) and, if alternative data samples were used to the model, this new model will fit the new sample equally well, but this may result in a high variance between the models, note that this depends on the sample size compared to the entire sample space. When the model has low complexity it is said to have high bias, this results in the model that represents function g fitting the sample data with high average error across all possible g (in the hypothesis space H) and the target function f , and so a high bias is conducive of a model that is insensitive to variations in training samples and will likely result in a high prediction error because the model fails to approximate f . The ideal model will therefore be a trade-off between bias and variance with the goal of achieving low bias and low variance (Friedman, 1997).

The penalty term in equation (4.15) is therefore designed to penalise complex models and λ is used to tune this penalty in an attempt to achieve the optimum bias and variance trade-off. A common regularisation function used in practice for the penalty term is the L_2 Norm of the model coefficients $\|\omega\|_2^2$. This is the sum of the square of the coefficients and forces a reduction of the

coefficient values that is spread across all ω without a reduction to zero (Ng, 2004). The L_1 norm $\|\omega\|_1$ is also a very common regularisation function and is the sum of the absolute values of the coefficients (Ng, 2004); an interesting side effect of this penalty is that it forces a reduction of the coefficient values that will tend to reduce coefficients to zero as it attempts to achieve a less complex model. This side effect can be useful for feature selection as the zero coefficients effectively eliminate features from the model and the remaining features are indicative of those features that were most important in discriminating between the target classes (Ng, 2004). Regularised Logistic Regression (RLR) has been investigated in (Vail & Veloso, 2008) for use in activity recognition by robots and by (Liu, et al., 2013) to inform sensor selection for HAR in ambient environments.

Decision Trees are another embedded method of feature selection where the selection of suitable features is an integrated element of the development of a tree (Wang & Li, 2008). Recall from chapter 3 that the ID3, C4.5 and the CART algorithms use information gain or gain ratio as a statistical test, in turn these heuristics require measures of uncertainty, often referred to as impurity (Alpaydin, 2014), that are associated with the class distribution of the data points before and after a split at a node in a decision tree. For example if N_m represents the number of data points at node m and if N_m^i represents the number of data points belonging to class label C^i at node m . Then the probability estimate for a class C^i is given by equation (4.16):

$$\hat{P}(C^i|x, m) = P_m^i = \frac{N_m^i}{N_m} \quad (4.16)$$

Hence if P_m^i for all i is 0, then none of the data points at node m are from the class C^i , and if it is 1 then all of the data points at node m are from the class C^i ; in each of these cases there is no uncertainty at node m and the node is said to be pure (Alpaydin, 2014). In between these extremes uncertainty is measured by various functions relating to the distribution of the target classes for a particular feature in the data set. The idea here is that all features relevant to a particular node should be tested and the feature which best discriminates between classes, with respect to the parent data, should be used to split the data points. An interesting side effect of this process is that it effectively ranks the features by their ability to classify the data at each node and

hence a form of feature selection is achieved whilst growing the DT. Entropy is one popular measure used to determine the information gain (Quinlan, 1986) at the nodes where equation (4.17) defines the entropy of the data at node m :

$$Entropy(m) = - \sum_{i=1}^k P_m^i \log_2 P_m^i \quad (4.17)$$

With the convention that $0 \log 0 = 0$ and k is the number of target classes. Each term in equation (4.17) is non-negative and close to 0 when P_m^i is either close to 0 or 1, hence the entropy will be small when most of the data in a node are from one class C^i and entropy is large if the data are spread across all classes C^1, \dots, C^k .

The Gini impurity shown in equation (4.18) is also a popular measure of impurity at a node and is a measure of inequality between the probability distributions of the target class labels.

$$Gini(m) = 1 - \sum_{i=1}^k P_m^i{}^2 \quad (4.18)$$

As with entropy the Gini impurity will be small when most of the data in a node belong to one class C^i and 0 indicating maximum purity. When uncertainty is maximised, at the point where all classes have equal representation, the Gini impurity is maximised at 0.5 indicating that the particular feature represented at the node is unable to discriminate between classes.

To determine how well the feature at each node discriminates between classes, a split is made for each feature and the difference between each impurity measure and the parent node is compared. The greater the difference between the nodes and the parent, the better the discriminative power of the feature. The gain (Δ) is calculated using equation (4.19):

$$\Delta = I(\text{parent}) - \sum_{i=1}^k \frac{N(m_i)}{N} I(m_i) \quad (4.19)$$

Where $I(\cdot)$ is the impurity measure of a node, N is the total number of data points at the parent

node, and $N(m_i)$ is the number of data points at child node m_i . When the impurity measure used in equation (4.19) is entropy, the Δ is referred to as information gain.

It can be proven that Entropy and Gini impurity measures tend to favour features that have a large number of possible values (Tan, et al., 2005), which may not necessarily be a reliable measure of the discriminatory power of the feature. The CART DT uses Gini impurity and is restricted to binary splits, to mitigate this issue, another method implemented in the C4.5 DT is to use gain ratio to calculate Δ :

$$Gain\ ratio = \frac{\Delta}{-\sum_{i=1}^k P(m_i) \log_2 P(m_i)} \quad (4.20)$$

Where the denominator is the split information and k is the total number of splits, therefore if a feature has a large number of values and hence produces a large number of splits the split information will be large and the gain ratio is reduced.

Embedded methods have the advantage of not needing to evaluate feature sets in an iterative process and are therefore computationally inexpensive with respect to wrapper methods (Alelyani, et al., 2013). They are specific to the machine learning algorithm and are therefore intrinsically tuned to the classification performance of the model used, often resulting in improved performance with respect to filtering techniques that are detached from the classifier (Liu, et al., 1996). Although the selected features from embedded methods are transparent and can be selected for use with unrelated, downstream learning algorithms there are no guarantees that these unrelated classifiers will produce comparable performance using these features (Chandrashekar & Sahin, 2014). It is for these reasons that a hybrid method (Das, 2001) that combines the embedded approach, along with subset selection using wrappers has been chosen here, This method matches the aim of identifying the subset of sensors that contribute most to the efficient classification of activity labels using classifiers that perform well on the specific data that is typically generated by ambient environments.

The empirical process described here is one which first determines the classifier or classifiers that produce the best classification of the data and so classifiers that incorporate an embedded feature

selection process have been used with the exception of the SVM with an RBF kernel. These models are then used to extract a list of sorted features, ranked by the importance to a particular model. The evaluation process uses K-fold cross-validation with backward selection over the ranked sensors. This feature ranking with backward selection forms the subset search heuristic over the feature space, by sorting the ranks from best to last it ensures that the subtracted features are the lower ranked features and so reduces the complexity to $O(N)$. The smallest subset that produces evaluation results comparable to the evaluation on the full subset is chosen as the subset of interest and can be analysed with the aim of evaluating the viability of the thesis hypothesis.

4.2 Algorithm Evaluation

The following section outlines the process of selecting machine learning classification algorithms using both the (Crandall & Cook, 2008) and (Tapia, et al., 2004) datasets, all experimentation was conducted using the Python, scikit-learn package (Pedregosa, et al., 2011). The purpose here is to empirically identify the most suitable algorithm, based on those highlighted in chapter 3 as the most popular in the field of HAR with a view to evaluation of the most appropriate feature selection method for this type of data.

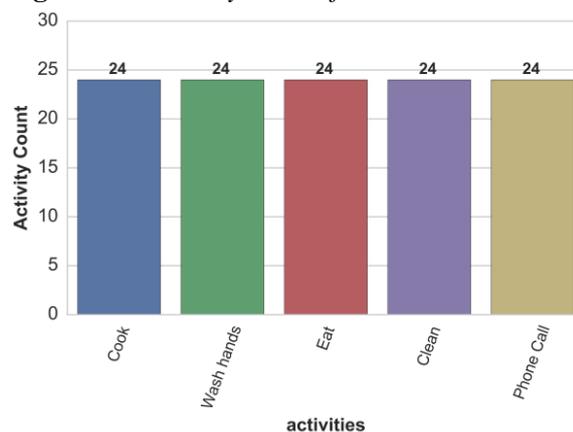
Each of the Machine learning algorithms discussed in chapter 3 have tuning parameters associated with them, for example the NB algorithm has the smoothing parameter alpha (α), used to account for features that may be absent during the learning process and LR has (λ), which is used to optimise the trade of between bias and variance. The values of these parameters are chosen to produce the most effective model during the training phase of the machine learning process. To facilitate this tuning of parameters the training set is split into training and validation sets, whereby the validation set is used to evaluate the model using suitable machine learning metrics also discussed in chapter 3. The test set is a portion of data that is held out and used to evaluate the final model chosen during the validation phase, this is often referred to as the hold-out set (Bishop, 2006), it is important that no data points from the test set are used through the validation process. In practice the datasets used in the HAR field are typically small and so splitting into Training and Test sets and further splitting the Training set into Training and Validation sets may mean that data points from the minority classes are not part of training or

validation sets. In such cases there may not be enough data points to create sufficiently generalised models to enable effective classification of out of sample data points. A technique known as cross-validation (Bishop, 2006), can be used to mitigate this issue by generating several (K is used to represent the actual number) training/validation set pairs from the original training set i.e. from a training set X , K training/validation set pairs $\{T_i, V_i\}_{i=1}^K$ can be used to determine model parameters. Those parameters producing the best results, averaged over the validation sets, can be chosen to perform the final training of a model (Alpaydin, 2014). As the datasets in HAR are typically unbalanced across the activity labels it is important that the activities are represented proportionally in both the training and validation sets, this is referred to as stratified cross validation (Kohavi & others, 1995) and is the process implemented throughout this thesis. It is worth pointing out that as K increases the number of data points used for training increases, but the validation sets are smaller. This is an issue when the data is not balanced across activity labels as there may be too few minority labels in the validation sets. A typical choice of K is 10 or 30 (Alpaydin, 2014) however this is determined by the class distribution across the dataset.

4.2.1 Algorithm Evaluation (Balanced dataset)

The (Crandall & Cook, 2008) dataset was used to evaluate NB, LR, SVM, RF and Adaboost classifiers. The graph shown in Figure 4.1 depicts the data as balanced across activity labels:

Figure 4.1: Activity counts for balanced dataset.



The dataset was divided into a training set and a test set using a stratified 70/30 split to maximise the activity class count in the test set. This produced a training set which was used to perform

stratified cross-validation to determine suitable model parameters for each classifier. To maintain continuity for comparison with the evaluation of the (Tapia, et al., 2004) dataset, and also because of the very limited number of data points, stratified cross fold validation was implemented with the common choice of K=10. A range of suitable values were tested for each of the parameters and the F1-measure was used to choose the best performing model and hence the parameters. The classifiers were then parameterised (with the chosen parameters), trained on the complete training set and tested on both the training set (this is to determine the fit of the model on the training set) and test sets, the results of which are shown in Table 4.1. It should be noted that as the dataset is balanced, the F1-measure will be similar to the measure of percentage accuracy of each model, and in this case either could have been chosen to evaluate the parameters, the F1-measure was chosen to enable comparison with the results of the unbalanced (Tapia, et al., 2004) dataset where the F1-measure is a more suitable metric.

Table 4.1 *Machine Learning classifier performance using balanced data set.*

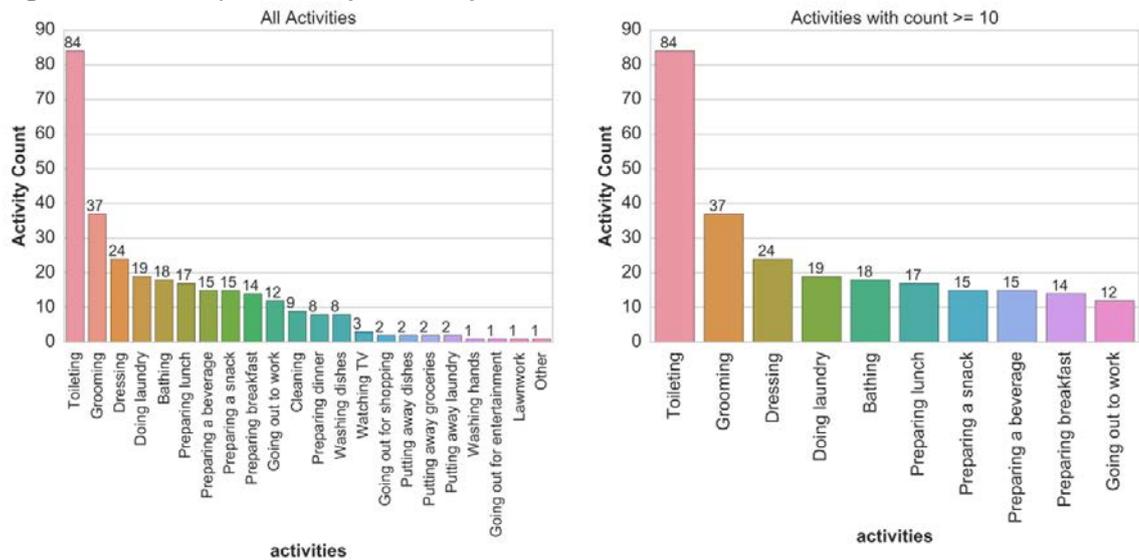
Machine Learning Model	Training set accuracy %	Test set accuracy %	Test set Precision %	Test set Recall %	Test set F1 score
NB (alpha=0.1)	95.24	97.14	97.50	97.14	97.13
SVM ((kernel=RBF, C=1.0)	100.00	74.28	83.14	74.29	75.02
SVM (kernel=linear, C=0.4)	100.00	100.00	100.00	100.00	100.00
LR (penalty = L1, C=0.4)	96.47	100.00	100.00	100.00	100.00
LR (penalty = L2, C=3.4)	100.00	100.00	100.00	100.00	100.00
DT (gini, depth=40)	100.00	97.14	97.50	97.14	97.13
DT (entropy, depth=20)	100.00	100.00	100.00	100.00	100.00
RF (gini, 15 learners, depth=40)	100.00	97.14	97.50	97.14	97.13
RF (entropy, 19 learners, depth=20)	100.00	100.00	100.00	100.00	100.00
Adaboost(gini, 12 learners, depth=10)	100.00	97.14	97.50	97.14	97.13
Adaboost(entropy,19 learners, depth=10)	100.00	100.00	100.00	100.00	100.00

The results show that for this dataset it is a trivial classification process because of the scripted non-stochastic, balanced nature of the dataset. Table 4.1 shows that the majority of classifiers perform extremely well with the exception of the SVM using an RBF kernel. In particular neither the RF, nor Adaboost classifiers perform any different to the associated DT classifier using the same criteria, indicating that neither randomisation, in the case of RF or using weightings in the case of Adaboost changes the model perspective of the data. The results indicate no real preference in the model to use for such data and hence any or all (with the exception of SVM-RBF) can be chosen to analyse the features of importance. Note: Precision, Recall and F1-measures have been scaled by a factor of 100 for clarity.

4.2.2 Algorithm Evaluation (Unbalanced dataset)

The (Tapia, et al., 2004) dataset was also used to evaluate NB, LR, SVM, RF and Adaboost classifiers. This dataset is not balanced across activity labels and this is illustrated in the class distribution bar graph shown to the left in Figure 4.2. The graph also shows that some activities have very few labelled instances, and so there is a danger that classification metrics may not clearly define the effectiveness of the model produced in learning (Maimon & Rokach, 2010). Also, as cross fold validation is being used for parameter evaluation there is a need to maintain sufficient labelled data from each class in the training set and test set, and so it follows that the original dataset should have sufficient members of each class to facilitate this. The Tapia dataset has several activities with very few data points and so it was necessary to discard those activities that show little representation and the dataset was reduced using a threshold of 10 class labels.

Figure 4.2: Activity counts before and after reduction.



This was followed by splitting the dataset using a 70/30 split into training and test sets. The algorithms were then used to perform stratified 8 fold cross validation, 8 folds were used as the lowest activity count was 12, when this is split using a stratified 70/30 split this leaves 8 data points for cross validation. The classifiers were then parameterised with the chosen parameters, trained on the complete training set and tested on both the training set and test sets, the results of which are shown in Table 4.2.

Table 4.2 Machine Learning classifier performance, unbalanced dataset

Machine Learning Model	Training set accuracy %	Test set accuracy %	Test set Precision %	Test set Recall %	Test set F1 score
NB (alpha=0.4)	88.20	67.53	70.10	63.63	65.87
SVM ((kernel=RBF, C=9.7)	93.82	76.62	83.33	73.07	73.87
SVM (kernel=linear, C=0.4)	93.26	77.92	84.63	76.34	77.68
LR (penalty = L1, C=1.9)	90.45	71.43	82.02	68.01	69.78
LR (penalty = L2, C=0.7)	89.33	71.43	80.08	69.61	72.14
DT (gini, depth=55)	99.44	64.94	60.31	59.16	58.81
DT (entropy, depth=59)	99.44	70.13	69.67	68.12	68.43
RF (gini, 86 learners, depth=55)	99.44	67.53	71.05	61.05	64.35
RF (entropy, 76 learners, depth=59)	99.44	67.53	77.21	62.07	65.02
Adaboost(gini, 97 learners, depth=10)	99.44	70.12	78.85	66.65	69.50
Adaboost(entropy,91 learners, depth=10)	99.44	68.83	73.42	64.92	66.82

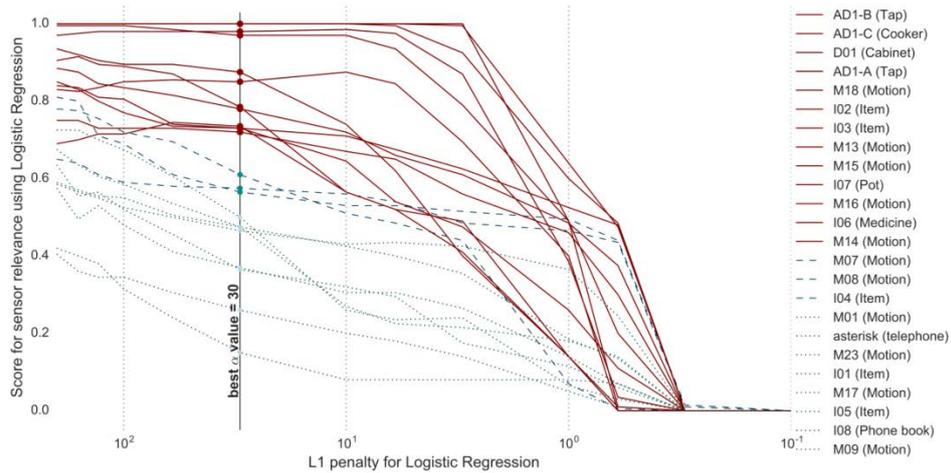
These results show that for this dataset the SVM classifiers outperform all others overall, followed by LR with an L2 penalty and then LR with an L1 penalty. When comparing these results with that of the balanced dataset it could suggest that classifiers based on the DT may perform better using data with a balanced distribution over the activity classes. The results in Table 4.2 show that the SVM with the linear kernel outperformed all other classifiers using this dataset, and so the coefficients can be examined to identify the most important sensors. This process is documented for both datasets in sections 4.3.1 and 4.3.2.

4.3 Sensor analysis using feature selection

The intuition behind embedded methods of feature selection can be illustrated using the (Cook & Schmitter-Edgecombe, 2009) dataset, by implementing a logistic regression algorithm whilst varying the value of the L_1 penalty using λ . Figure 4.3 shows the graph produced using a Randomised Logistic Regression (RLR) model, from the python scikit-learn (Pedregosa, et al., 2011) package, on the CASAS dataset. The LR algorithm was used to train several models with the following values of a parameter $C \in \{200,160,130,100,60,30,10,6,3,1,0.6,0.3,0.1,0.06,0.03,0.01\}$

Note: In the scikit-learn RLR model implementation, the parameter C is used to tune the L1 penalty where C is the inverse of λ , therefore as C decreases the coefficients of the model are forced to decrease producing a less complex, higher bias model. In RLR the coefficient values ω can be used to assess the contribution of the feature to the resultant model and in the scikit-learn RLR model these coefficients are normalised to provide a score between 0 and 1 for each feature.

Figure 4.3 Logistic Regression with L1 penalty, balanced data set.



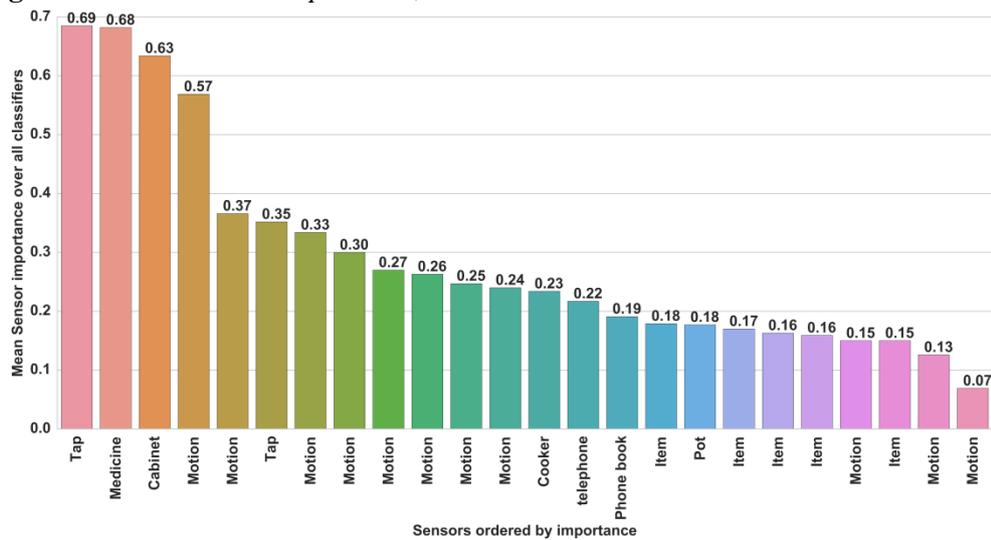
This score can in turn be used to represent the importance of the features to the model. As models with high variance are sensitive to changes in the data sample, only using one sample of the training data for evaluation would not be stable, by taking many samples and evaluating the coefficients and hence the importance of the feature across all samples, a more stable score is achieved, this is also the purpose of cross validation. The scikit-learn RLR model can be used to resample a training set a number of times (in this case the data was resampled 200 times) create a new model for each resample, evaluate the model and score the features for that model. The maximum score for each feature over the 200 samples is selected and the overall rank is depicted in the key on the right of Figure 4.3.

To produce Figure 4.3 the RLR model was used to extract the selected feature scores over the range of C values, with a view to illustrating the concept of the reducing coefficient contribution as the L1 penalty increases, ultimately reducing all feature contributions to zero as the bias of the resulting model increases beyond the point of reasonable representation of the data. The vertical line in the graph represents the C value that produces the most accurate model taken from a 10 fold cross validated grid search using the same range of C values. The solid lines represent the 13 (out of 24) sensors that contribute most to the accuracy of the model, at the point where $C = 30$; $\lambda = 1/30$. It is interesting to note here that 62% of these sensors relate to motion, water taps or electrical appliances. Here the cross validation grid search was set to use accuracy as a metric as the CASAS dataset is well balanced across the target classes as was illustrated in the previous section.

4.3.1 Feature selection analysis (Balanced dataset)

Using the results from the balanced dataset shown in Table 4.1, section 4.2.1 a judgement on the best models to use for feature, and thus sensor analysis can be undertaken. Table 4.1 indicates that all classifiers performed well on this dataset, with the exception of the SVM using the RBF kernel. As feature selection is an embedded part of each of these algorithms and the scikit-learn (Pedregosa, et al., 2011) package provides a mechanism to extract the features in terms of an importance ranking, one method of evaluating the importance of sensors over a range of models is to take the mean importance rank across all those models that performed well, Figure 4.4 shows the mean sensor importance rank across all models, disregarding the SVM with RBF kernel.

Figure 4.4 Mean Sensor Importance, balanced dataset.



It is again encouraging to note that from analysis of the mean ranks shown in Figure 4.4 only 1 sensor of type item (medicine) scores in the top 60%, with the majority of the top 60% being motion sensors followed by taps, a cabinet, the cooker and telephone. To re-evaluate the classifiers, the top 60% of sensors were extracted by thresholding the rank at a value of 0.2. The data was modified to include only the extracted features and the process described in section 4.2.1 was repeated with results shown in Table 4.3:

Table 4.3 Classifier performance using sensor subset, balanced dataset.

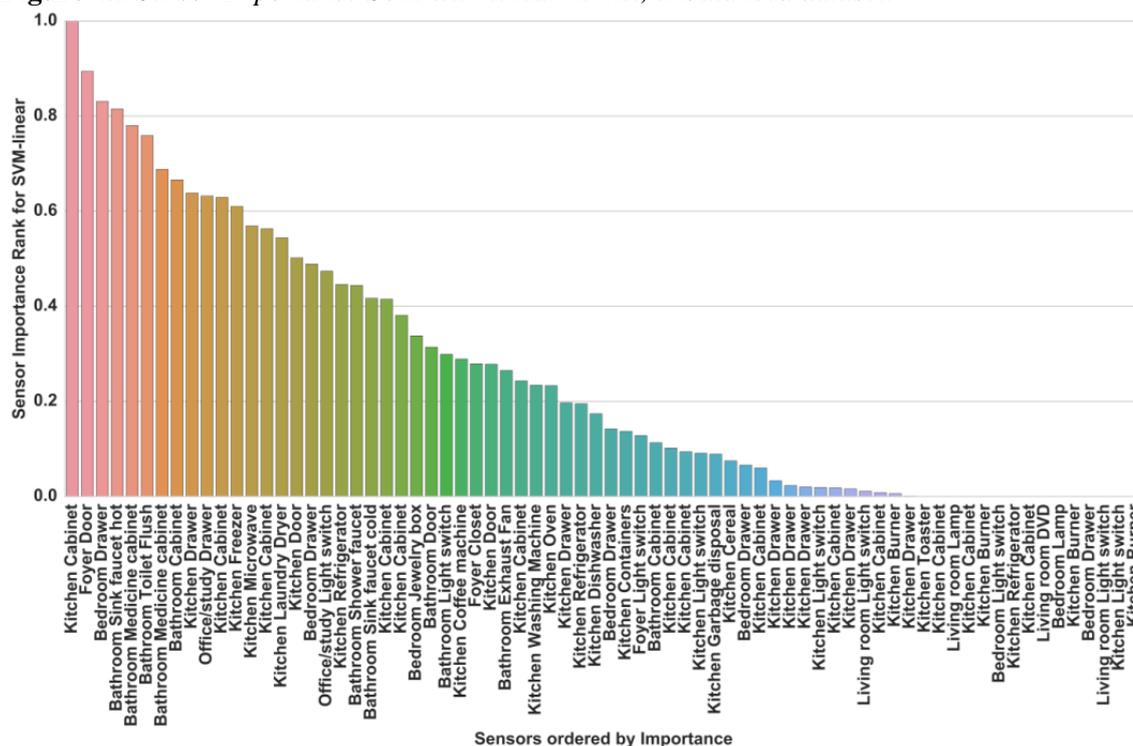
Machine Learning Model	Training set accuracy %	Test set accuracy %	Test set Precision %	Test set Recall %	Test set F1
NB (alpha=0.1)	91.76	97.14	97.50	97.14	97.13
SVM (kernel=linear, C=0.3)	100.00	91.43	91.67	91.43	91.38
LR (penalty = L1, C=0.4)	95.29	91.43	94.00	91.43	91.60
LR (penalty = L2, C=0.2)	95.29	94.29	95.56	94.29	94.42
DT (gini, depth=51)	100.00	91.43	92.14	91.43	91.60
DT (entropy, depth=60)	100.00	94.29	94.29	94.29	94.29
RF (gini, 26 learners, depth=51)	100.00	94.29	94.29	94.29	94.29
RF (entropy, 29 learners, depth=60)	100.00	94.29	94.29	94.29	94.29
Adaboost(gini, 53 learners, depth=10)	100.00	91.43	92.14	91.43	91.60
Adaboost(entropy,19 learners, depth=10)	100.00	91.43	91.79	91.43	91.41

The results indicate that most classifiers lost performance with the exception of NB, which maintained performance with the sensor subset. Although the other classifiers were not as effective they still show very good performance on the data using the sensor subset and the results encourage the thesis hypothesis that it may be possible to replace individually mounted sensors with PIR sensors along with water and electrical disaggregation and with careful selection of the classification algorithm achieve high performance on out of sample data. Here the thresholding of the feature ranks was evaluated by observation; this is in line with the aims in this chapter and part of the analysis process of the thesis but is ultimately an impractical approach for feature selection. The following section outlines a wrapper evaluation method that uses backward selection of the sorted ranked feature list producing a hybrid embedded/wrapper selection method for HAR data.

4.3.2 Feature selection analysis (Unbalanced dataset)

The results in Table 4.2 show that the SVM with linear kernel has a better performance on the (Tapia, et al., 2004) dataset with an F1-measure of 77.68. Only 2 other classifiers performed with F1-measures above 70, the SVM with RBF kernel and the LR classifier using the L2 penalty each providing F1-measures of 73.87 and 72.14 respectively. With this in mind it is justifiable to consider the feature importance of the SVM with a linear kernel and evaluate this classifier on the ranked subset of features. Using the scikit-learn package the rank of each feature was extracted from the SVM-linear model and Figure 4.5 illustrates these ranks using a bar graph sorted by feature importance.

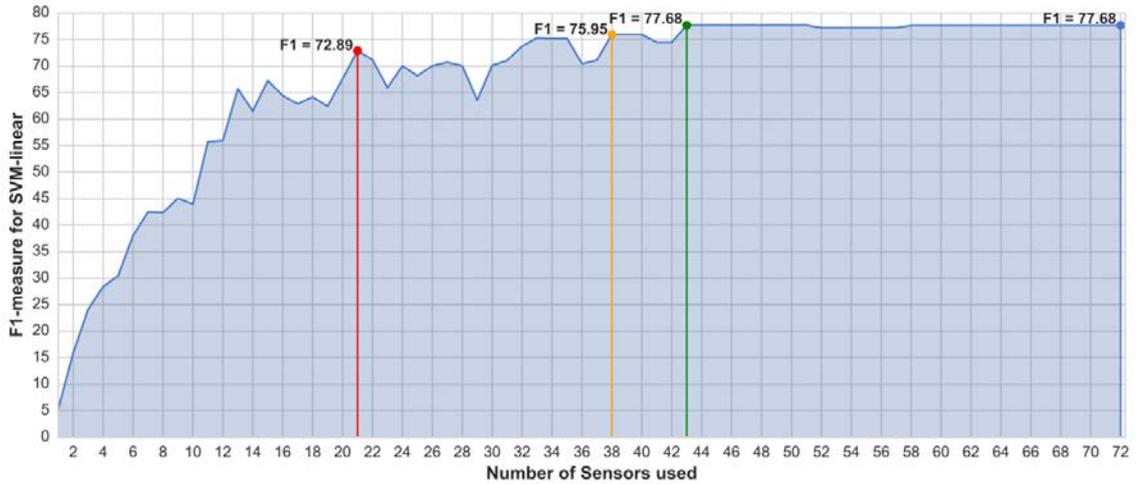
Figure 4.5 Sensor Importance SVM with linear kernel, unbalanced dataset.



The graph shows that, although several sensor that rank highly were affixed to cabinets, drawers and doors; of the top 60% (43) ranked sensors, 18 can be attributed to electricity or water usage and out of the remaining 25 it can be argued that their importance may be due to locality, that is a specific activity e.g. preparing breakfast in the kitchen may place high importance on the “Kitchen Cabinet” sensor. As this dataset does not use PIR sensors it is proposed that it may be possible to replace such sensors with PIR sensors and maintain locality information that may render cabinet, draw and door sensors redundant.

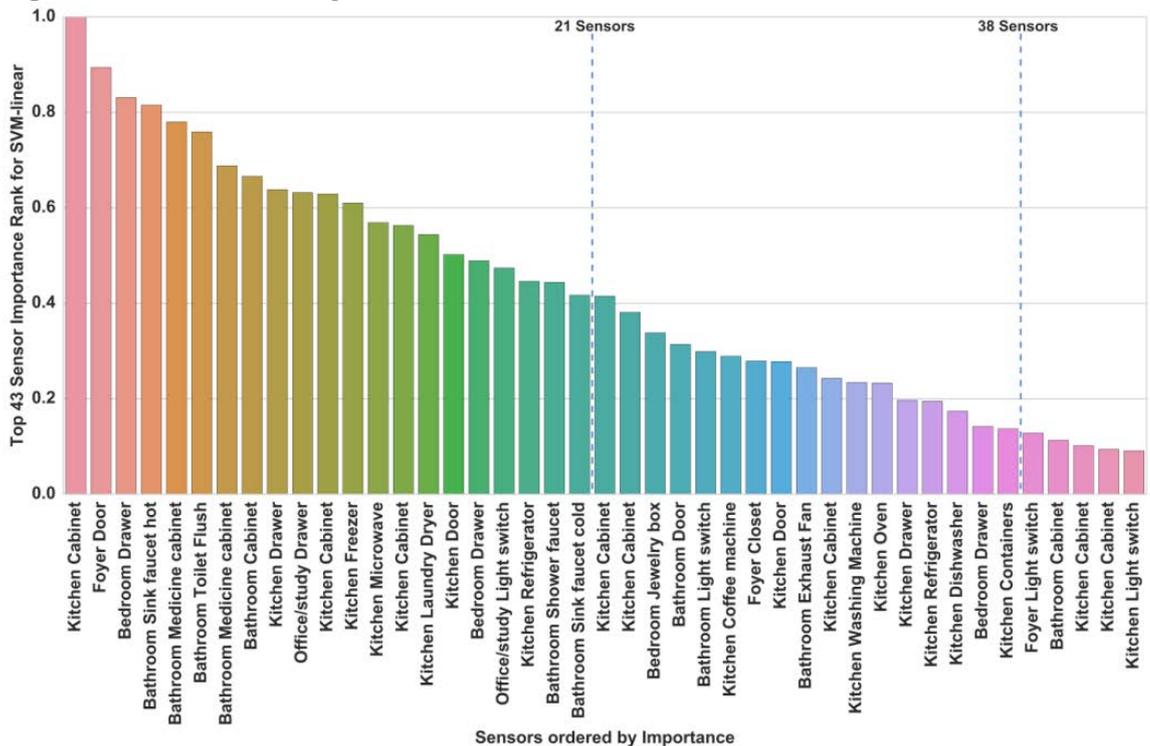
In order to determine the comparative accuracy of the SVM-linear model using a feature when compared to not using that feature, the model was retrained and evaluated using subsets of the ranked sensors. This was achieved by first selecting all 72 sensors, using validation to determine C, parameterising and testing, then removing the sensor with least ranked importance and repeating on that subset. The F1-measure was recorded for each model trained and tested using an ever reducing subset of sensors and the graph of Figure 4.6 was produced. The graph shows that using the SVM with linear kernel and ranking the sensors by importance it is possible to use a subset of the top ranking 43 sensors with no reduction in the F1-measure of the classifier.

Figure 4.6 Sensor subset Performance SVM with linear kernel, unbalanced dataset.



The graph also highlights that an F1-measure of 72.89 is achievable with a subset of just 20 sensors from the original 72 available. Figure 4.7 shows the top 43 ranked sensors, identifying the cut-off point for the top 38 and top 21 ranked sensors.

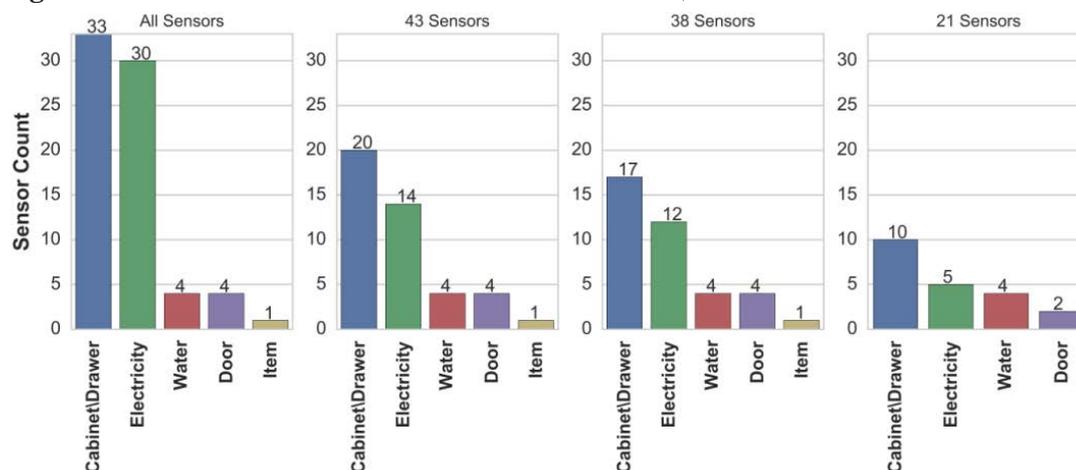
Figure 4.7 Sensor subset importance, unbalanced dataset.



It is noteworthy that those sensors that relate directly to water and the use of doors persist throughout the subsets; this is significant as the aim of this work is to establish whether water usage and locality (through the use of PIR sensors) play a significant role in activity recognition. A significant number of sensors relating to electricity are present in the top 38 although these

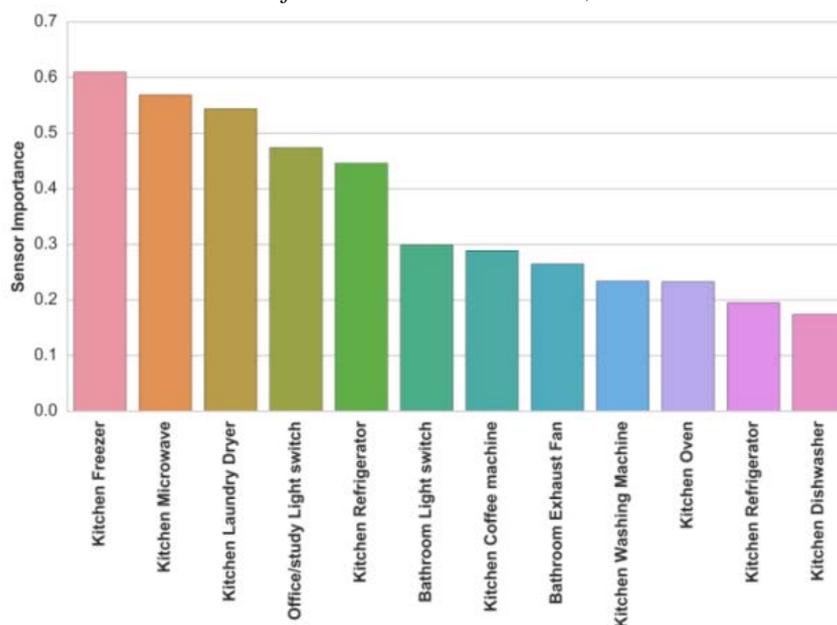
have reduced significantly from the full sensor set and this may point to quite a few electrical sensors that could be redundant in the classification process. Figure 4.8 shows the sensor counts for the various subsets grouped into types of “Cabinet/Draw”, “Electricity”, “Water”, “Door” and “Item” (Note: the item sensor is a single kitchen container).

Figure 4.8 Sensor subset counts - SVM with linear kernel, unbalanced dataset.



Further examination of the sensors that relate to electrical devices can be undertaken to determine those devices that are most important for the recognition of activities of interest and may help in the decision making process when developing activity recognition systems that use electrical disaggregation. Out of the 38 sub-sensor grouping, the 12 sensors that relate to electricity usage are shown in Figure 4.9 and are ordered by the importance rank:

Figure 4.9 Electrical sensors from the 38 sensor subset, unbalanced dataset.



The electrical devices may provide a hint as to the activity of interest e.g. “Kitchen Freezer”, “Kitchen Microwave” may indicate “Preparation of food” or “Cooking” activities and so encourage the inclusion of electrical disaggregation in the thesis hypothesis. Based on this analysis the proposal of this thesis to replace all water usage and electrical appliance sensors with a fusion of water and electrical disaggregation is encouraged. It is envisaged that this technique will facilitate a more practical system of activity recognition that can utilise modified version of devices that already exist in these environments i.e. smart meters and alarm systems.

4.4 Summary and Conclusion of Feature Selection for Activity Recognition

This chapter was dedicated to the analysis of two data sets with a view to determine those sensors that contribute most to the performance of Machine Learning classification models that perform well on the publically available, Human Activity datasets. The concept of feature selection has been discussed and definitions of the three common feature selection methods of; Filter (Liu, et al., 1996), Wrapper (Kohavi & John, 1997) and embedded (Neumann, et al., 2005) have been introduced. These methods have been reviewed in terms of appropriate use for sensor selection and the hybrid feature selection method proven to be most appropriate. The hybrid method proposed has been described and can be summarised as implementing embedded selection through algorithm performance evaluation, extraction of sorted, ranked features followed by wrapper subset selection using backward elimination of the lowest ranking features. A detailed discussion on embedded methods has been included to create intuition and justify its use here, including some minor experimentation and illustrations to emphasise the concepts.

Six classification algorithms, chosen because they produce ranked features, have been evaluated using the scikit-learn package (Pedregosa, et al., 2011) on both the scripted, balanced (Cook & Schmitter-Edgecombe, 2009) dataset and the unscripted unbalanced (Tapia, et al., 2004) dataset, the results of which are produced in Table 4.1 and Table 4.2 respectively. The performance results using the balanced dataset did not produce a standout classification algorithm so a mean ranking across all classifier feature ranks (excluding SVM with RBF kernel) was produced and the algorithms re-evaluated using 10-fold cross validation and 60% of the top ranked features. The algorithms were then re-evaluated using a test set and showed a performance reduction in all

algorithms, but no algorithm produced accuracy below 90% or F1-measure below 90 on the test set. More significantly the feature selection highlighted the majority of sensor types in the top 60% were associated with locality, water or electricity usage.

The performance results using the unbalanced dataset showed the SVM with a linear kernel as having the best performance overall and so the features were extracted and sorted by rank. These features were iteratively evaluated using wrapper feature selection, the SVM with linear kernel and backward feature elimination, reducing the subset by 1, low ranking feature at each iteration. The visualisation in Figure 4.7 shows the F1-measure of classifier evaluation against sensor count, for each iteration, and confirms that it is possible to reduce the number of sensors, in this dataset from 72 to 43 with no reduction in classifier performance. Moreover the reduced sensor subset shows a large proportion of electrical appliances, no reduction in water usage sensors or locality sensors, and so goes some way to confirming the hypothesis of this thesis.

Chapter 5 Repurposed Smart Meters

Many of the studies carried out in human activity recognition, for example (Lee & Mase, 2002), (Philipose, et al., 2004), (Steinhauer, et al., 2010), (Logan, et al., 2007) and (Tapia, et al., 2004) focus on the use of ubiquitous sensitised spaces or video based surveillance for the source of sensor data. These techniques suffer from many challenges that have been highlighted in chapter 2, the most notable of which are in the areas of privacy and the difficulty of collecting sufficient labelled data. These methods are therefore, relatively intrusive and often require expensive computational input as well as a high level of deployment impracticality. Chapter 4 described an investigation into the importance of the individual sensors used in ambient environments that implement large numbers of sensors, for the purpose of human activity recognition. Research carried out using feature selection on two publicly available datasets concluded that it is a viable option to investigate the possibility of only considering sensors that produce signals that represent water utility, electrical appliance usage and human movement for the classification of key activities of interest. In addition to this it is proposed here that using disaggregated data from repurposed smart meters could be a viable alternative to what is often perceived as intrusive recognition technology that may result in a reluctance to accept large quantities of sensors in practical environments (Garg, et al., 2014). Methods of disaggregation at the electricity and water meter points have proven to perform exceptionally well when trained with large quantities of data (Chen, et al., 2005), (Fogarty, et al., 2006), (Fogarty, et al., 2006), but gathering and labelling this data is, in itself, an intrusive process that requires significant effort and could compromise the practicality of such promising systems. In this chapter an investigation of the viability of implementing utility meter disaggregation is investigated and an attempt to mitigate the issue of insufficient labelled data using data synthesis on a water meter disaggregation system is described. This is a significant contribution because by using very few reference data points, the practicality of such systems compared to those currently requiring significant training and calibration times can be significantly improved. This is in addition to proving the viability of water and electricity meter disaggregation as a method for activity recognition that would reduce the complexity of the practical implementation into retrofit environments.

5.1 Using a water meter as sensors

It is proposed in this chapter that a system of non-intrusive sensor fusion that implements disaggregation techniques, at both the water (Srinivasan, et al., 2011) and electricity meters (Gupta et al., 2010), may go some way to reducing the quantity of sensors required in an ambient environment to enable the recognition of activities of daily living. Chapter 4 highlighted the observation that many of the sensors that were selected as important for the purpose of activity classification may be important to indicate locality and could therefore be replaced with PIR sensors. The inclusion of PIR sensors in the environment can replace large numbers of sensors that provide locality data, and also complement the disaggregated data from each utility meter. This contextual data from the PIR sensors is used to overcome the problem of similar signatures created by similar devices, by indicating the location of the device in use. For example two similar toilets will have similar water use signatures but will very rarely be in the same room. This section documents the results of testing three different supervised machine learning techniques on water flow data gathered from an ambient environment, using a prototype water meter with a simple Hall Effect water flow sensor and PIR sensors located in the rooms. This data is used to train and test Artificial Neural Network (ANN), Support Vector Machine (SVM) and K-Nearest Neighbour (KNN) classifiers, using validation over various model parameters, to determine the performance of each method using labelled data to provide a baseline for comparison followed by models trained with synthesised data. A method of training data synthesis is proposed and described in detail. A comparison of the resulting classification performance of the models, first with the classifiers trained with labelled data generated from a purpose built smart water meter, and then using synthesised training data created from data points that are carefully selected from the original data is undertaken.

5.1.1 Related Work

In the area of water utility usage monitoring as an activity recognition sensor the authors of (Chen, et al., 2005) carried out a study on automated bathroom activity monitoring using sound signals from microphones fitted in a small shower room. Hidden Markov models (HMM) were

used as the classification technique and produced an average classification accuracy of 83.55% over six activity types. There is an acknowledged issue of privacy, whether actual or perceived, in this work but the results are indeed encouraging. To produce the models used in the real trials, the authors used a training period of 10 days, representing a major practical limitation of the system. Research carried out by (Fogarty, et al., 2006) extended the idea of using microphones for activity recognition beyond the bathroom by fitting devices to the cold and hot water inlet as well as the main soil stack. This study reports an average recognition accuracy of 86.75% over 8 water usage event types. This technique alleviates the privacy issues highlighted in (Chen, et al., 2005) by taking the microphones out of the private areas and into the basement (the location of inlets and outlets), but still suffers from the need for extensive labelled training data to create the classification models. The study also acknowledges a reliability issue when classifying events generated from similar fixtures, as well as issues relating to ambient noise pick-up on the microphones.

A system proposed by (Froehlich, et al., 2009) that replaces the microphones of (Chen, et al., 2005) and (Fogarty, et al., 2006) with a pressure sensor used to classify the pressure changes associated with the opening and closing of valves at individual fixture events. This method resolves the issues of privacy by replacing microphones with pressure sensors, and is also less intrusive as the sensors can be fitted to external water taps. The study reports results of 97.9% using a template classification model combined with a KNN classifier, but also acknowledges the deficiencies in classification of compound events, however the publication does not indicate the training data requirements of the system. The study was followed up by (Larson, et al., 2012) with an extended analysis of the technique as well as a comparison of the template model against an HMM classifier. This follow up study reports that both techniques produce aggregate classification accuracy of greater than 90%, but confirms that significant calibration on the training data is necessary and there still remain some limitations in the classification of compound events.

Work carried out by (Srinivasan, et al., 2011) solved the problem of classifying similar events in different locations by using data from PIR sensors to determine context. The study focus was

primarily on water disaggregation to determine fixture level consumption, and reported 86% recognition accuracy over 467 fixture events. The work used a combination of Canny Edge detection and unsupervised Bayesian inference and reported that no training data is required, but the fixture level identification process requires very specific flow rates and event durations and thus suggests the possible need for extensive calibration. The idea of using PIR sensors to determine context is documented in chapter 6 of this thesis when used during final experimentation.

The development of an automatic flow trace analysis system was proposed by (Nguyen, et al., 2013a) and a classification model that uses a Dynamic Time Warping (DTW) algorithm and HMM's along with event probability data to identify single fixture level events. The study uses labelled data gathered from 252 residential households over a period of 2 years during the summer and winter months. The published paper reports an average classification accuracy of 84.1% and was compared to a commercial package that takes the water efficiency and flow rate statistics of each fixture to determine usage. This study was followed up by (Nguyen, et al., 2013b) where the issue of classifying combined fixture level events is presented and tackled by incorporating gradient vector filtering to separate combined fixture level events into single events before using the HMM models for event classification. The study reports that approximately 88% of combined fixture level events were classified accurately, and also suggests that future research would require the collection and manual labelling of a much larger sample of combined events. A recent publication by (Nguyen, et al., 2014) presents an adaptive system that can use the initial classification model learned from the 252 residential households and adapts this for new previously unseen households. The basic principle of the system is to use the current model to determine any fixture level events that cannot be classified, these events are then classified and added to the model and used to improve system performance. The study also discussed the development of a software package which, as presented, allows the occupants to manually modify the labelled data and upload this to the database to improve classification accuracy. Using this system the study reports that most of the achieved recognition accuracies for all end use categories were approximately 90%.

5.1.2 Synthesising labelled training data

Like many other machine learning applications, all of the discussed methods of disaggregation require many samples of labelled training data to perform efficiently, but the practicalities of collecting this labelled data in a domestic environment are restricted by both time and access constraints. These restrictions on labelled data collection time and access can also result in labelled data that is unbalanced in favour of data point classes that occur more frequently, such as was shown in Chapters 3 and 4 with the unbalanced (Tapia, et al., 2004) dataset, and hence can result in biased machine learning models. Synthesis of training data for machine learning applications suffering from a shortage of initial labelled training data has been used in emotion recognition techniques (Schuller & Burkhardt, 2010), where speech is synthesised to train acoustic models for emotional speech recognition. The researchers compared models trained with a select set of emotional speech patterns from a database of human emotional speech and a synthesised version of these. The paper concludes that in cases where the models are trained using data synthesised using statistical methods, these models can outperform those trained with non-synthesised human emotional speech data.

The use of synthesised training data has been widely used in the field of hand-written text recognition originally by (Ha & Bunke, 1997) and followed up by (Varga & Bunke, 2003) who published work that used synthesised hand written text, generated from the distortion of real lines of text, to train an HMM-based handwritten sentence recognizer resulting in an overall performance improvement in the detection of handwritten sentences. The principle of interpolating synthetic data from real world data within the bounds of a well-defined domain has been used in a wide range of fields such as the detection of Mine-Like objects from Sonar imagery (Barngrover, et al., 2015), the testing of uniaxial compressive strength (UCS) of rocks in engineering geology (Sezer, et al., 2014) and image classification to enable Unmanned Aerial Vehicles (UAVs) to see and avoid each other (Rozantsev, et al., 2015).

5.1.3 Comparison of ANN, SVM and KNN algorithms

ANN and SVM classification techniques have been compared in many studies over a diverse topical range from drug/non drug compound filtering (Byvatov, et al., 2003), computer security (Chen, et al., 2005) to classification of images in image retrieval systems (Wong & Hsu, 2006). In all cases the SVM model outperforms the ANN classifier, however the reasons for this are not clear from the studies, but research has shown (Hable & Christmann, 2011) that support vector machines perform very well when there are significant errors in a small fraction of the data set, as well as when there are small errors spread across the whole data set. There is no available data to indicate that the ANN should be disregarded as a possible classifier for use in event disaggregation at the water meter, so this model is considered here. The promising results produced by (Froehlich, et al., 2009) using a template model combined with a KNN classifier have inspired the inclusion of a KNN classifier here, although the template model has been omitted due to the need for calibration of the template with training data.

5.1.4 Artificial Neural Networks (ANN)

An ANN classifier (Gardner & Dorling, 1998) is composed of a number of interconnected nodes that represent a mapping between the inputs to an initial (input) layer and the outputs from an output (classification) layer of nodes. The network can include any number of internal (or hidden) layers of nodes, with each node representing a non-linear activation function that is modified by a weight. Various algorithms are used to determine the values of the weights for a given set of training examples. In the case of the back-propagation algorithm, commonly used with ANN's, the feedback from the output layer is generated by comparing the output with labelled examples consisting of input and labelled output, and propagating the error back through the network to modify the weight of the internal activation functions. The network is considered trained, and generalisation achieved, at the point of minimum error in the output layer across all training examples. Once trained, new unlabelled examples are presented to the input layer and fed forward through the network producing a classification at the output layer. The memory requirements for an ANN classifier during test are proportional to the number of weights in the

network and so are small in comparison to a KNN classifier, however training an ANN can be slow due to the nature of the back-propagation algorithm and is proportional to the dimensionality of the data and the number of iterations required to minimise the error at the output. The dimensionality of the training data in this study, up to one hundred features depending on the final parameters, suggests that the ANN may be relatively slow to train, however it is included here because of the low memory requirements once trained.

5.1.5 Support Vector Machine (SVM)

As described in chapter 3 an SVM classifier (Boser, et al., 1992) uses a linear hyperplane, determined by labelled training data, to identify the separation between classes of examples. The hyperplane is chosen in such a way as to maximise the margin between the two classes and minimize the generalisation error. A function determined by a selection of support vectors, and representing the chosen hyperplane can be chosen to modify the feature space in cases where the classifications of data points are not a clear linear separation. Chapter 3 also describes the SVM as a quadratic optimization problem and thus has a high algorithmic complexity with the memory requirements proportional to the number of training examples. This implementation uses training examples made up of up to 100 water samples each of which represent a feature and thus a dimension in SVM terms, this moderate dimensional space is a suitable application domain for an SVM classifier and because the focus here is on the synthesis of training data, that may inherently exhibit a high generalization error, the SVM classifier is used to mitigate against this. The most basic SVM is useful when the data to be classified is linearly separable i.e. there exists a linear decision surface or hyperplane that can separate the classes without error. The linear function (Boser, et al., 1992) for SVM classifiers is given by equation 5.21 :

$$f(\vec{x}) = \text{sign}(\vec{w}^T \vec{x} + b) = \quad (5.21)$$

Where a function return value of -1 indicates one class and a return value of +1 the other class. For multiple class problems, the method most often used is that of one versus all classification (Statnikov, et al., 2011), this process uses (k) binary SVM classifiers to separate k classes. For

example if $k = 3$, the first SVM would separate class 1 from a combination of classes 2 and 3, the second SVM would attempt to separate class 2 from a combination of classes 1 and 3 and the final SVM would attempt to separate class 3 from a combination of classes 1 and 2. New unclassified data points are then evaluated using all SVM models and a voting system that assigns a value based on the distance from the hyperplane for all +1 values returned from the models (Statnikov, et al., 2011). When the data is not linearly separable the data used in the classifier can be mapped from the input feature space \vec{x} to another feature space \vec{f} using a non-linear function $\Phi: \vec{x} \rightarrow \vec{f}$:

$$f(\vec{x}) = \vec{w}^T \Phi(\vec{x}) + b \quad (5.22)$$

Where Φ is a function that maps the data into the new feature space (Ben-hur & Weston, 2007). Kernel methods however are implemented to represent the data slightly differently, and use pairs of stepwise comparisons instead of a mapping function. This way a dataset can be represented by a real valued pairwise comparison function $k: X \times X \rightarrow \mathbb{R}$ and instead of the mapping $\Phi: \vec{x} \rightarrow \vec{f}$, the dataset is represented by an $n \times n$ matrix of pairwise comparisons $k_{i,j} = k(x_i, x_j)$ (Vert, et al., 2004).

Several kernels exist that are popular in the field with the next up step from linear being the polynomial kernel. The feature space for polynomial kernels consists of all monomials up to the degree (d) of the kernel, for example for \vec{x} of length m , the feature space would be transformed to: $x_1^{d_1}, x_2^{d_2} \dots x_m^{d_m}$, with $d = 1$ the equivalent to the linear SVM. The Radial basis function (RBF) kernel is also very common in the field and is defined as:

$$k(x, \acute{x}) = \exp\left(-\frac{\|x-\acute{x}\|}{2\sigma^2}\right) \quad (5.23)$$

All three kernels are implemented here to evaluate the performance of each and choose a suitable classifier for further evaluation.

5.1.6 K-Nearest Neighbours (KNN)

The KNN classifier (Cover & Hart, 1967) uses a distance metric comparison between known, labelled data points in the dataset and new incoming data points. The classification of new data point x is achieved by using a number (k) of the nearest labelled examples to vote on the classification of x :

$$Y(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} Y_i \quad (5.24)$$

Where $N_k(x)$ is the neighbourhood of x with the k nearest data points in the labelled dataset. The memory requirement for a KNN classifier, at test time, is large as the set of the stored labelled examples, must be available to perform the comparison, however the computational requirements of the distance metric calculation is small compared to that of the SVM and ANN classifiers. The most common distance metric used is that of Euclidean distance, which is often appropriate for quantitative features and is simply the straight line distance between the data points in Euclidean space. Here Euclidean distance is compared to standardised Euclidean distance where each data point coordinate is scaled by the standard deviation of all data points in the labelled training data.

5.1.7 The problem with labelled training data

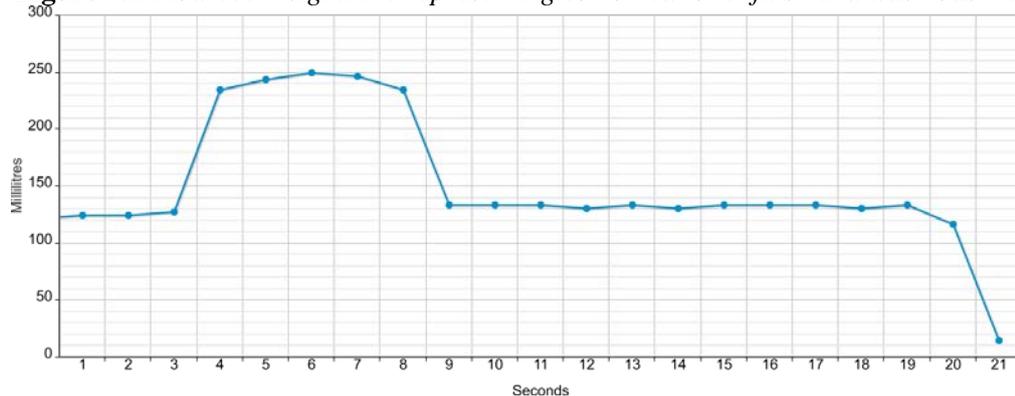
For each of these Machine learning classifier models there is a training requirement for relatively large labelled data sets. This may have a significant impact on the practicality and feasibility of using machine learning techniques to train such system for use in the field. To mitigate this issue a method of training data synthesis is proposed and as such the focus and unique contribution of this chapter is to show that, by using domain knowledge and a simple distortion algorithm, large labelled data sets can be synthesised. These data sets can then be used to train models that perform comparably to the water disaggregation systems used in (Chen, et al., 2005), (Fogarty, et al., 2006), (Froehlich, et al., 2009), (Srinivasan, et al., 2011), (Larson, et al., 2012), (Nguyen, et al., 2013a), (Nguyen, et al., 2013b) and (Nguyen, et al., 2014). By using a small amount of labelled training data, that represents the extremes of fixture usage within the ambient environment, a large quantity of simulated data representing the many possibilities between these

extremes, can be generated and used to train the machine learning algorithms. The particular domain of water meter disaggregation serves as a suitable test case for labelled data simulation as the extremes of data flow are limited by the practical application of the fixtures, for example a standard water closet (WC) has a flush mechanism that is limited by the maximum capacity of the cistern, and likewise a wash basin or bath have a limited maximum capacity and useful daily functionality can be assumed with a fair degree of accuracy.

5.2 Evaluation of Machine Learning Algorithms for water meter Disaggregation

A prototype water meter was developed and implemented to record the flow rate of water through a mains inlet, cold water feed pipe. The meter was programmed and calibrated to sample in millilitres per second (ml/s), capturing data on the rising edge, of an event, from zero ml/s and completing the capture on the falling edge back to zero ml/s. The meter was fitted with an interface to an ambient environment with a core KNX (Richardson, 2015) infrastructure and a KNX mobile phone application was configured to send a signal from the user to the water meter to indicate the ground truth label of a particular event. An example of data captured for a compound toilet and wash basin event is shown in Figure 5.1, where each point represents the millilitres of water flow in a second.

Figure 5.1 Flow event signature representing combined toilet flush and wash basin usage.



To prove the concept and provide a focus for classifier comparison this experimentation is restricted to a single room equipped with toilet, wash basin, and shower. Restricting the study to a single room, at this stage, removes any ambiguity caused by similar fixtures situated in other locations and also removes the need for PIR data. The final system, as discussed in chapter 6,

uses a classifier that is trained on each room, and PIR data will determine which classifier should be considered to determine the final classification. Data was captured in a real en-suite environment occupied by a middle aged couple for a period of two months using the prototype water meter. The event signatures were analysed, by hand, and after removal of ambiguous data, a total of 306 events remained. Table 5.1 shows the separation of these events into specific event types along with the quantity of each.

Table 5.1 *Event name and quantity of clean events captured*

Event Type	Class	Quantity
Toilet	1	73
Wash basin	2	157
Shower	3	56
Toilet + Wash basin	4	5
Shower + Toilet	5	15

These events were used to train, and test, the three types of classifier discussed in the previous section, a KNN, one versus all SVM and an ANN using a single hidden layer and the back-propagation algorithm. Table 5.1 shows that the number of class 4 and 5 training data points was small in comparison to classes 1, 2 and 3, showing an imbalance in the data set and so highlighting the limitations of such systems when labelled training data is limited. The classifiers were trained and tested using a variation of feature vector lengths. The lowest feature vector length of 3 was implemented using the average flow (ml/s), the standard deviation of flow and the total volume as the feature set. To obtain a feature set, of length greater than three, the water flow rate samples representing flow rate in ml/s were manipulated, with samples that represent possibly redundant data disregarded. This redundant data was characterised by areas of a particular training data point that did not change over time and thus may not provide any additional information relevant to classification. Using this technique, the length of the feature set for each labelled example could be varied over the full range of samples. For example to obtain a labelled data point with 10 features, the most frequently occurring sample value (mode) over the entire feature set of the data point was determined, and one such sample equalling this mode value was extracted. This process was then repeated until the data point was reduced to 10

sample points creating a feature vector length of 10; if a data point had less than 10 sample points then zeros were added to the feature vector padding the vector to a length of 10. This technique was used to create separate training sets with feature vector lengths of 10, 30, 60 and 100, thus along with the data points consisting of 3-features a representative range of datasets with feature vector lengths between 3 and 100 was produced.

Although the longest water meter feature vector in the labelled data set consisted of 376 samples (representing a shower event), a feature count of 100 was chosen as the upper limit. This makes sense when the labelled data are analysed as, from observation, the majority of the 376 samples represent possible redundant features, which are characterised as samples that are repetitive. This is illustrated in Table 5.2, which shows the breakdown of the 376 samples of a shower event data point before and after a feature resize to 100 features.

Table 5.2 *Sample values that make up the majority of the 376 features of a shower event.*

Sample value from event	# before resize	# after resize
266	135	14
269	76	14
263	63	14
272	54	14
275	18	14
# with other sample values	30	30

The table shows that the original labelled data point includes 135 samples that have a recorded flow rate of 266 ml/s (row one of Table 5.2) and it is argued here that a very simple method of feature reduction can be used to eliminate duplicate features in the data to obtain the feature length of 100, with the caveat that the same technique is used during live event classification. In the resize process the table indicates that the 135 samples recording a flow rate of 266 ml/s are reduced to 14 in number. The repeated extraction of the sample with the most frequently occurring flow rate is used to remove redundant data and the labelled data point can be represented using feature lengths that are smaller than the original. Table 5.2 also shows that the majority of data is restricted to 5 flow rate values that are relatively similar and can be reduced in quantity while maintaining the general shape of the event, and thus the mutual information between event data point and classification label. After feature reduction, a process of parameter

optimisation was carried out by validating each model using 2-fold cross validation. Cross validation with more than 2 folds i.e. K-folds, could have been used but the small number of class 4 examples is a restriction on K and with 2 folds we can guarantee at least 2 data points of class 4 events in the training set. The standard parameters for each model, discussed previously were varied and tested to determine the performance of the models and provided a best in class for each of the ANN, SVM and KNN techniques.

To determine the most appropriate SVM kernel to represent the SVM classification technique, the following kernels were trained (Boser, et al., 1992); linear function, (RBF), Polynomial with orders 2, 3 and 4 and the sigmoid function. These classifiers were tested using 2-fold cross validation and 100 features. Each SVM model was tested for parameter selection with the best parameters producing the results shown in Table 5.3. It is acknowledged here that classification accuracy can be an erroneous metric due to the imbalance of classes in the data set, but it is used here in comparison over identical training and validation sets, purely for parameter selection and comparison with later experimentation on a balanced dataset.

Table 5.3 SVM kernel accuracy using 2-fold cross validation and 100 features

SVM kernel type	C (soft-margin)	Accuracy (%)
Linear	1.00	74.83
2nd Degree Polynomial	0.18	80.13
3rd Degree Polynomial	0.18	79.47
4th Degree Polynomial	0.03	79.47
Radial basis Function(RBF)	5.66	84.77

The results show that the RBF kernel performed considerably better than any of the alternatives with an accuracy of 84.77% a soft margin of 5.66 and 100 features, not shown in the table is the gamma value of $2.2e^{-5}$ that was used to achieve this performance.

An ANN (Gardner & Dorling, 1998) was designed using a single hidden layer and trained using 100 features, with values of the learning rate (lambda) ranging from 0 to 1 in steps of 0.3 and a range of hidden nodes from 100 to 500. These results are illustrated in Table 5.4, which shows a maximum accuracy of 82.37% with 160 hidden nodes and a lambda of 0.3.

Table 5.4 ANN accuracy using 2-fold cross validation and 100 features

# Hidden Nodes	Lambda			
	1	0.6	0.3	0
100	81.41%	81.41%	81.41%	82.05%
130	81.09%	81.41%	80.77%	81.41%
160	80.77%	81.73%	82.37%	81.73%
190	80.45%	81.09%	81.73%	81.73%
300	80.77%	81.73%	80.77%	80.45%
400	80.45%	81.09%	81.73%	81.41%
500	81.09%	81.73%	81.09%	81.09%

A KNN (Cover & Hart, 1967) classifier was used to classify events with 100 features with a range of nearest neighbours and using both Euclidean and SEuclidean distance, Table 5.5 shows the results and highlights a best accuracy of 86.09%.

Table 5.5 KNN Accuracy using 2-fold cross validation and 100 features

# Neighbours	Euclidean	SEuclidean
1	85.10%	86.09%
2	84.11%	84.11%
3	85.10%	86.09%
4	85.10%	84.77%
5	81.46%	81.13%

These preliminary results show that the accuracy of the KNN classifier is marginally better than that of the SVM or ANN trained on unbalanced real data using 100 features and 2-fold cross validation. Further tests were carried out with feature vector sizes of 3, 10, 25, 30, 45 and 60. The results obtained for varying feature vector sizes were analysed and the best performing models are summarised in Table 5.6, where the accuracy column displays the classification accuracy of the best in class models.

Table 5.6 Summary of best performing model trained and tested with real data

Model	# Features	Recall (class 4)	Recall (class 5)	Accuracy
SVM(RBF)	25	0.00	0.00	89.07%
ANN(160 nodes)	60	0.00	0.00	83.01%
1 – NN	30	0.50	0.71	89.40%

It must be noted here that neither the SVM or ANN models were able to detect class 4 or class 5 events during testing, whereas the KNN model detected 1/2 and 5/7 respectively, this is recall and is shown in columns 3 and 4 of Table 5.6. This inability to recall class 4 and 5 events, highlights the issues related to the limited number of training examples with ANN and SVM classifiers. In this case there were only 2 training data points for class 4 events and 7 for class 5 events. This represents a small proportion of the overall number of training data points and thus they have limited impact on the overall accuracy scores. The complete precision, recall and F1 graphs using real training data are shown in Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5. Note: Precision, Recall and F1, as in chapter 4, have been scaled by a factor of 100 for clarity and where no bar is shown a value of zero was recorded and replaces the bar in the figure.

Figure 5.2 Precision chart for SVM, ANN and KNN classifiers

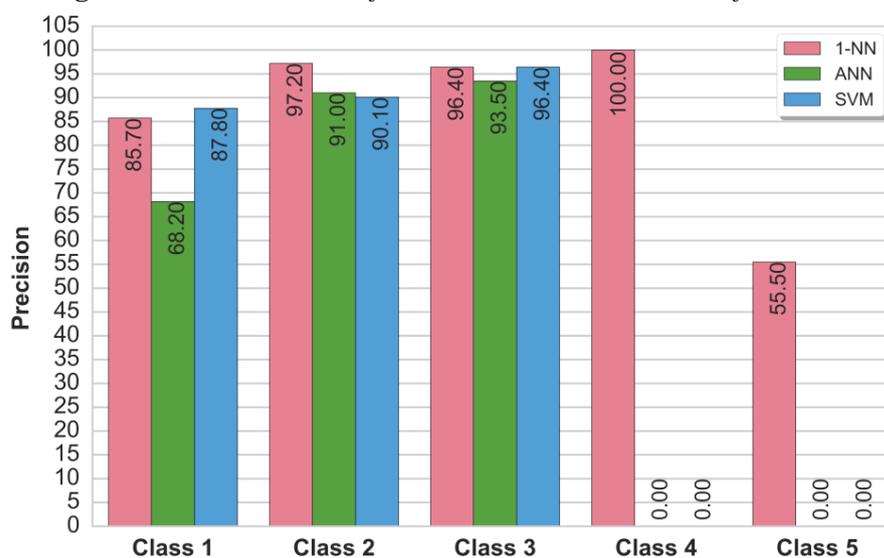


Table 5.2 shows that the 1-NN classifier had the best precision scores overall and especially for class 4 and 5 data points with the ANN and SVM classifiers producing zero precision for these class labels suggesting that the low number of class 4 and 5 labels effect the SVM and ANN classifiers where data points from other classes are mistaken for these.

Figure 5.3 Recall chart for SVM, ANN and KNN classifiers

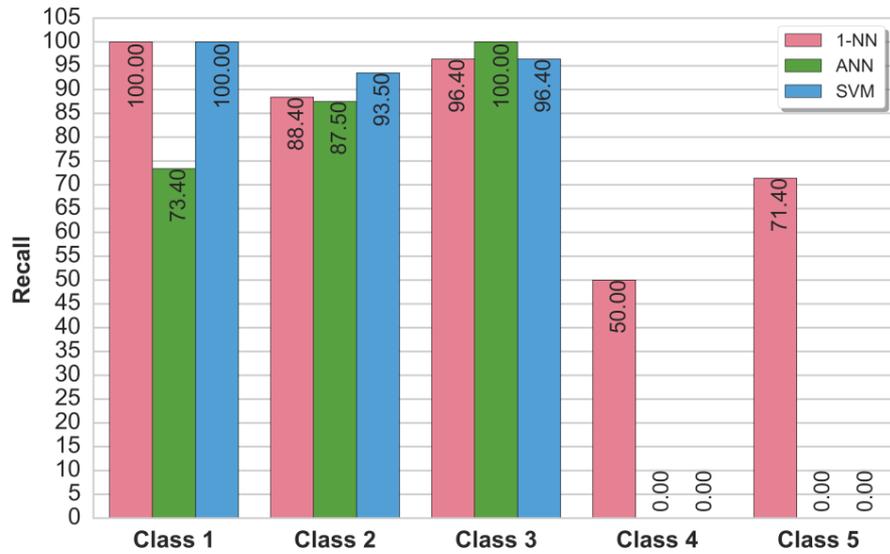


Figure 5.3 shows that the ANN and SVM classifiers had zero recall scores for class 4 and 5 data points suggesting that they struggled to identify these classes. All classifiers struggled to identify the class 4 labels in the dataset.

Figure 5.4 F1 chart for SVM, ANN and KNN classifiers

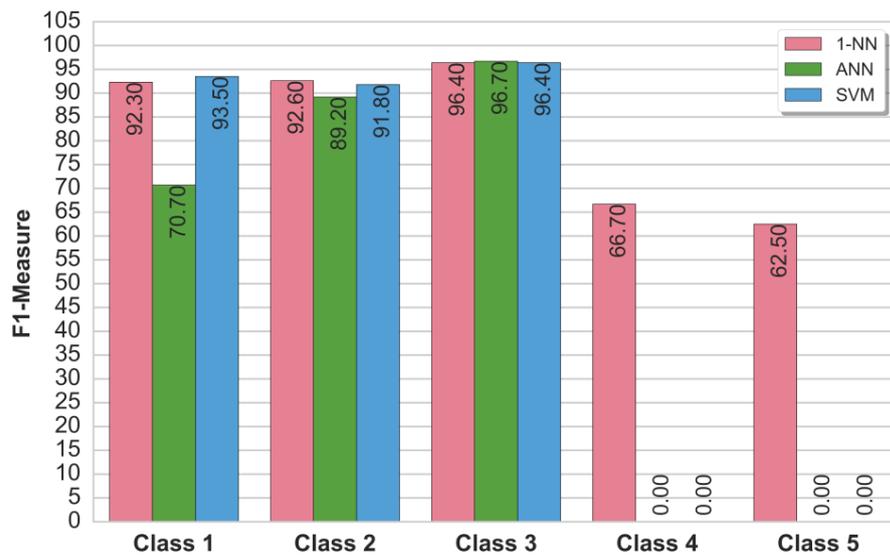
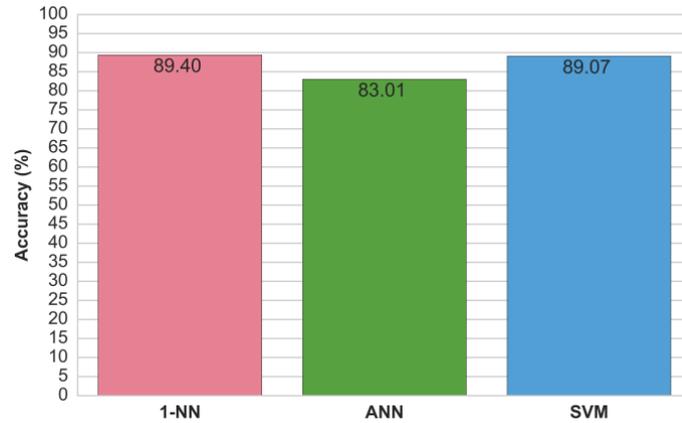


Figure 5.4 shows the F1 measures over each class and provides a composite view of Figure 5.2 and Figure 5.3, confirming the inferior performance of the ANN and SVM classifiers over classes 4 and 5 and the superior performance of the KNN classifier. The overall accuracy of the models is shown in Figure 5.5.

Figure 5.5 Accuracy chart for SVM, ANN and KNN classifiers



5.3 Learning with synthesised training data

This chapter is primarily focused on the practical limitations on collecting suitable training data that this method of disaggregation at the meter point relies upon. For such a system to be practical it is necessary to label training data that is representative of the application environment. As stated this would involve a prolonged period of user training that may be unacceptable in a real environment; these issues are evident in (Chen, et al., 2005), (Fogarty, et al., 2006), (Froehlich, et al., 2009), (Srinivasan, et al., 2011), (Larson, et al., 2012), (Nguyen, et al., 2013a), (Nguyen, et al., 2013b) and for the initial data collection in (Nguyen, et al., 2014). To mitigate this, a process of artificial data synthesis was used to generate a range of labelled training data points, which were used to determine if training the models on synthesised data would produce comparable results to that of the real data.

To synthesise the data, two labelled data points were chosen from the real data set, as representative of the extremes of an event class. For example, a toilet flush had data points that consisted of 15 and 60 samples at the minimum and maximum, respectively. The synthesised data were created by stretching the data point with the minimum water flow samples, until the number of samples matched that of the data point with the maximum water flow samples. As each new sample was added to the original, a new data point was recorded for that event, thus event data points representing a full range of toilet flushes were synthesised.

Algorithm 1: Simulating data from two selected examples of none compound events

input : $E_{(min)}\{x_0, \dots, x_{min}\}, E_{(max)}\{x_0, \dots, x_{max}\}, x_h, x_t$

output: A series of synthesised training examples $E_{(s)}$

determine $E_{(max)}\{x_h, \dots, x_t\} \in E_{(max)}\{x_0, \dots, x_{max}\};$

while $x! = x_t$ **do**

$x_m \leftarrow \text{Mode} (E_{(max)}\{x_h, \dots, x_t\});$
 $x_{max} \leftarrow \text{MaxValue} (E_{(max)}\{x_h, \dots, x_t\});$
 $x_{min} \leftarrow \text{MinValue} (E_{(max)}\{x_h, \dots, x_t\});$
 $x_{noise} \leftarrow \text{Random} (0, x_{max} - x_{min});$
 $x_{new} \leftarrow x_m + x_{noise};$
 $E_{(sx)}\{x_0, \dots, x_n\} \leftarrow E_{(min)}\{x_0, \dots, x_{new}, x_t, \dots, x_{min}\};$
 $E_{(sfx)} \leftarrow \text{FlipHorizontal} (E_{(sx)});$
 Save $E_{(sx)};$
 Save $E_{(sfx)};$

$E_{(s)} \leftarrow E_{(sx)} + E_{(sfx)};$

Algorithm 1 outlines this process. This was then repeated in reverse from maximum to minimum, a process not shown in Algorithm 1 but uses a similar technique, however instead of adding samples, extracts the mode sample from the event. Each new event was then flipped horizontally to create a mirror image version and saved as a labelled data point. For events with multiple components i.e. Toilet + Wash basin, the proportion of the event that represent the combined samples was shifted up and down the event, synthesising new data points with each shift as well as recording horizontally flipped versions, Algorithm 2 outlines this process.

Algorithm 2: Simulating data from a single example of compound a event

```
input :  $E\{x_0, \dots, x_n\}, x_h, x_t, x_{cb}, x_{ce}$   
output: A series of synthesised training examples  $E_{(s)}$   
  
determine  $E\{x_h, \dots, x_t\} \in E\{x_0, \dots, x_n\}$ ;  
determine  $E\{x_{cb}, \dots, x_{ce}\} \in E\{x_h, \dots, x_t\}$ ;  
while  $x_{ce} \neq x_n$  do  
     $x_m \leftarrow \text{Mode}(E\{x_t, \dots, x_{cb-1}\})$ ;  
     $x_{max} \leftarrow \text{MaxValue}(E\{x_t, \dots, x_{cb-1}\})$ ;  
     $x_{min} \leftarrow \text{MinValue}(E\{x_t, \dots, x_{cb-1}\})$ ;  
     $x_{noise} \leftarrow \text{Random}(0, x_{max} - x_{min})$ ;  
     $x_{new} \leftarrow x_m + x_{noise}$ ;  
     $E_{(s)}\{x_0, \dots, x_n\} \leftarrow E\{x_0, \dots, x_{new}, x_{cb}, \dots, x_{ce}, \dots, x_t, \dots, x_n\}$ ;  
     $E_{(sf)} \leftarrow \text{FlipHorizontal}(E_{(s)})$ ;  
    Save  $E_{(s)}$ ;  
    Save  $E_{(sf)}$ ;  
     $E\{x_0, \dots, x_n\} \leftarrow E_{(s)}$ ;  
    determine  $E\{x_h, \dots, x_t\} \in E\{x_0, \dots, x_n\}$ ;  
    determine  $E\{x_{cb}, \dots, x_{ce}\} \in E\{x_h, \dots, x_t\}$ ;  
  
 $E_{(s)} \leftarrow E_{(s)} + E_{(sf)}$ ;
```

Table 5.7 shows the number of original (real) events compared with the simulated event data points produced using this technique.

Table 5.7 Event type with the quantity of original real events, and the quantity of synthesised events synthesised from only 2 of the extreme data points from originals.

Event type	# real	# synthesised
Toilet	73	262
Wash basin	157	330
Shower	56	157
Toilet + Wash basin	5	166
Shower + Toilet	15	233

The synthesised data points were then used to determine the parameters for the models by using 2-fold cross validation. Although the number of class 4 and 5 data points has been significantly increased, the use of 2-fold cross validation is maintained for comparison purposes. The models were then parameterised and trained using the synthesised data and tested using the real data.

5.4 Experimental Results and Discussion

To determine whether the problems associated with scarcity of training data can be improved upon using synthesised data, the experimentation was repeated using data produced using the methods described in the previous section. Table 5.8 shows the best performing models trained with the synthesised data and tested with the real data (this was previously used for training and testing).

Table 5.8 Summary of the best performing model trained with synthesised data and tested with real data.

Model	# Features	Recall (class 4)	Recall (class 5)	Accuracy
SVM(RBF)	3	1.00	1.00	83.33%
ANN(100 nodes)	30	1.00	1.00	79.09%
1 – NN	40	1.00	1.00	84.97%

The results show that by using synthesised data, the recall measure for class 4 and class 5 events has improved to the point of total recall. The complete precision, recall, F1 and accuracy graphs using synthesised training data are shown in Figure 5.6, Figure 5.7, Figure 5.8 and Figure 5.9.

Note: Again Precision, Recall and F1 have been scaled by a factor of 100 for clarity.

Figure 5.6 Precision chart using synthesised training data.

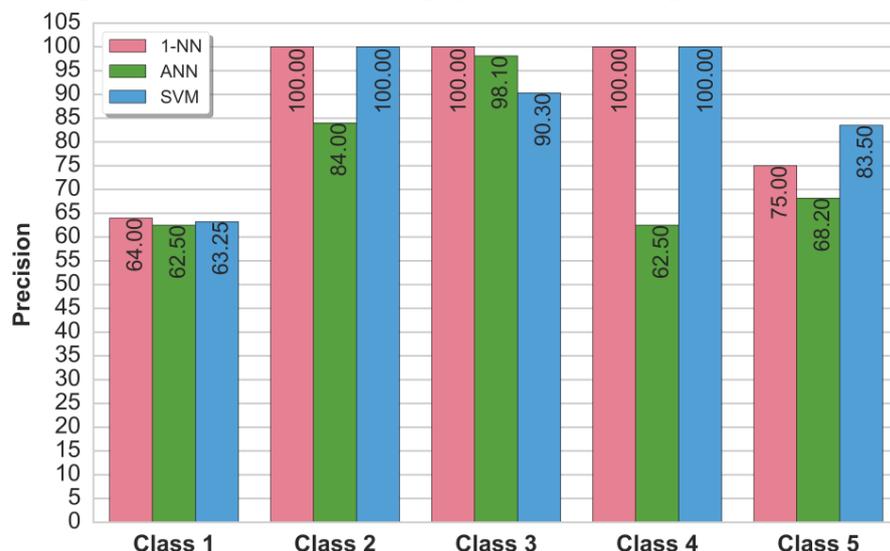


Figure 5.6 shows that although the precision for each classifier has reduced for class 1 labels there is a significant performance increase in precision for class 4 and 5 data points suggesting that the synthesised data has enabled the classifiers to differentiate better between these and other

data points. The precision of the KNN classifier has improved marginally over class 2, 3 and 5 data points but the SVM classifier precision has improved significantly using synthesised training data.

Figure 5.7 Recall chart using synthesised training data.

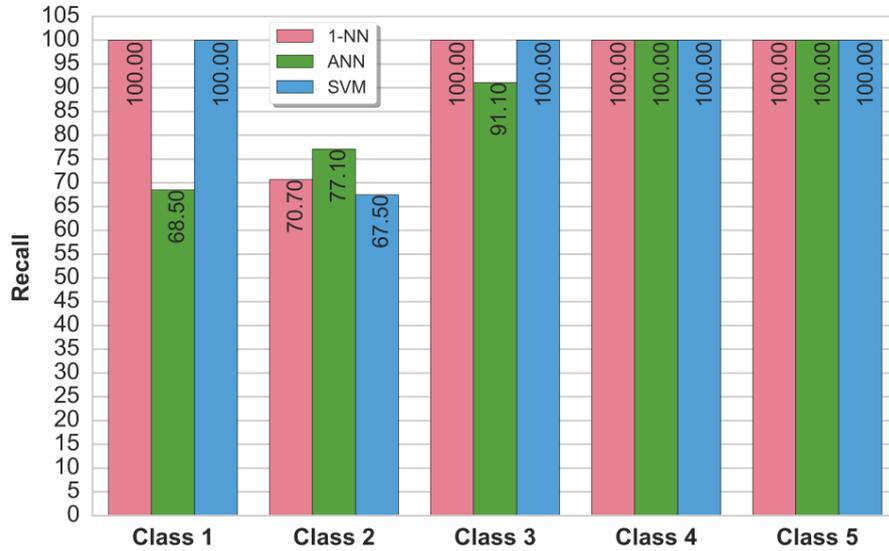


Figure 5.7 shows that the ANN classifier has lost recall marginally on the class 1, 2 and 3 data points but provides total recall using synthesised training data. The SVM and KNN classifiers have lost recall on class 2 data points but again have improved to total recall on classes 4 and 5.

Figure 5.8 F1 chart using synthesised training data.

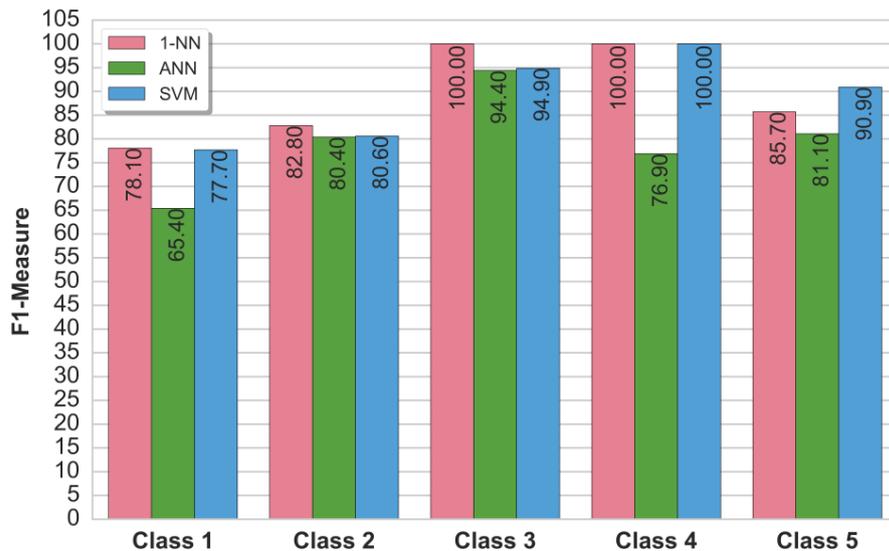
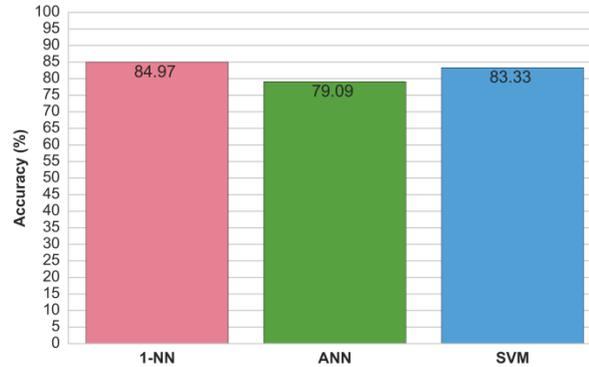


Figure 5.8 uses the F1 measures to summarise the improvement of the classifiers and although there is a reduction in performance for class 1 and 2 events this is balanced by a significant

improvement in performance over each of the other classes for each model. Overall the accuracy of each model using synthesised training data has reduced marginally but this underlines the problem using accuracy as a classifier metric when the data is unbalanced. Figure 5.9 shows the accuracy for each of the models using synthesised training data.

Figure 5.9 Accuracy chart for synthesised training data.

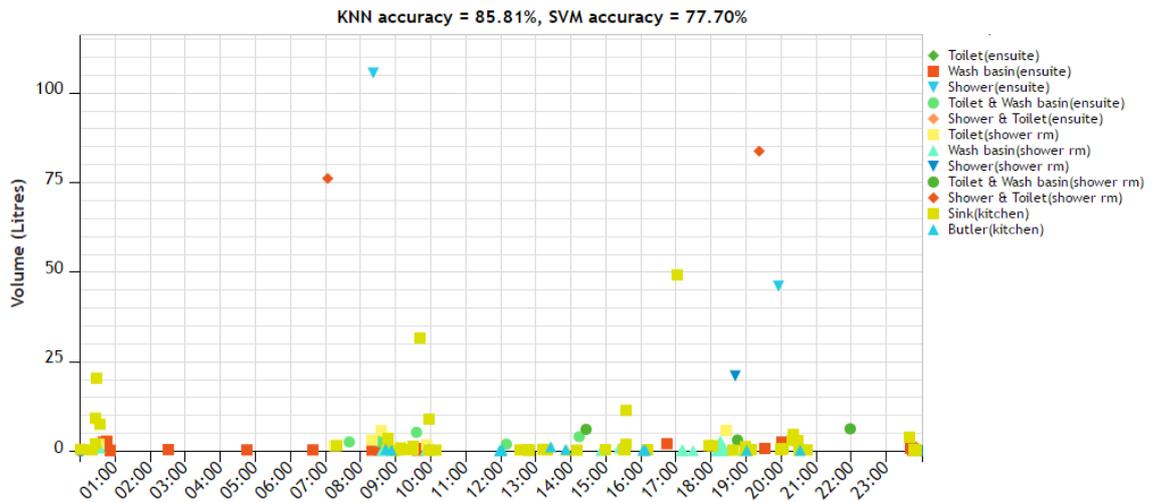


In summary, it has been shown that in cases where the quantity of labelled training data points are limited, as was the case with class 4 and 5 events, the SVM and ANN classifiers were unable to classify these events at test time; this was not the case with the KNN classifier. It has also been shown that by artificially synthesising labelled data that are representative of the differences between two extremes of each real data class, accuracies of 83.33%, 79.09% and 84.97% with the SVM, ANN and KNN classifiers, respectively can be achieved. These accuracies are slightly lower than the accuracies of 89.07%, 83.01% and 89.4% achieved without synthesis but, the measure of accuracy masks the underlying imbalance of representative classes in the data set. When metrics of precision and recall are used to produce an average F1 score for each model the improvements using synthesised data become apparent.

The system described here was augmented with PIR sensors and expanded to include the kitchen and shower room as well as the en-suite, of a 3 bedroom ambient environment (more details of this environment are provided in chapter 6). Inspired by the results shown in Figure 5.6, Figure 5.7 and Figure 5.8 the KNN and SVM classifiers were used as well as an online tagging system developed to monitor live performance per day, this is shown in Figure 5.10 were the results for 28/05/2016 are shown. The key of Figure 5.10 shows the fixture events that were monitored for the three rooms and live accuracy measures are provided. A process of automated teaching of the

water meter was implemented by development of a calibration mode where the user is able to teach the meter with a series of water usage extremes for each event. This facilitates the use of longer or shorter training periods to produce more or less labelled training data for synthesis. Algorithm 1 and 2 were implemented to synthesise the labelled data of single and compound events respectively and the chart of Figure 5.10 was produced using only 2 representative water meter data points per event, and so represents a worse case training scenario. Future work will be carried out to gauge the optimum quantity of labelled data points required before synthesis.

Figure 5.10 Screen shot from online water disaggregation meter



It should be noted here that the water meter data represented in the chart of Figure 5.10 was generated whilst the environment was occupied by four individuals. In chapter 6 the data is cleaned to enable analysis of data produced by one individual, this is justifiable and in-line with the thesis hypothesis as the aims here are to determine whether activity recognition for single occupancy environments is possible using water and electricity usage and movement data.

Nonintrusive appliance load monitoring is a process or system that monitors the electrical appliances on a standard electrical circuit, either commercial or residential. These systems are designed to infer the usage statistics of an individual device from a central location, often the point of metering (Hart, 1992). The process is more commonly referred to as disaggregation at the electric meter point (Kim, et al., 2009). Electricity meter disaggregation has been demonstrated successfully in (Patel, et al., 2007) and (Gupta, et al., 2010) with the research conducted by the former reporting the ability to detect the on/off events from low powered

appliances such as televisions, laptops, refrigerator door lights and individual room lights as well as higher power events such as an Electric Oven. A detailed research study and practical evaluation of disaggregation at the electricity meter is beyond the scope of this thesis but in the same way that repurposing the water meter to produce water usage events based on appliance usage has been shown to provide useful, non-intrusive, event data. It can be argued that by repurposing the electricity meter to one which can detect and identify the operation of key electrical appliances within the environment, it may also be possible to provide non-intrusive event data for recognition of activities such as “Cooking”, “Hot Beverage”, “Watching T.V.” or “Ironing” based on detecting on/off events from the Oven, microwave, hob, kettle, T.V and Iron. The ambient environment that has been created and fitted with the smart water meter, as detailed in this chapter, has also been fitted with selected individual device electric meters, the specifics of which will be deferred to chapter 6. The data from these meters will be used as events to physically simulate a disaggregation process with the development of a custom built smart electricity meter left as future work.

5.5 Summary and Conclusion of Repurposed smart meters

This chapter has presented a method of water disaggregation at the meter point using a purpose built Hall Effect water meter and has compared the classification accuracy of ANN, SVM and KNN classifiers on predicting the fixture responsible for an event, using both labelled data collected over a two month period and synthesised data generated from only two extremes of labelled data per event class. This is a significant contribution as for such a system to be practical, the implementation and training times of the disaggregation system must also be practical. It has been shown that by using synthesised data, automatically generated by algorithms developed from knowledge of the domain, that training times for water meter disaggregation can be reduced to minutes rather than hours or days in some cases (Srinivasan, et al., 2011). An accuracy of 84.97%, is comparable to that of (Chen, et al., 2005), (Fogarty, et al., 2006), (Froehlich, et al., 2009), (Srinivasan, et al., 2011), (Larson, et al., 2012), (Nguyen, et al., 2013a), and (Nguyen, et al., 2013b), however these systems rely on real data for training and calibration resulting in significant limitations in terms of time and access to the environment

under study. The accuracy of 84.97% produced here should be considered worst case as only 2 labelled data point examples, representing water fixture event extremes, are used to generate the synthesised training data and more example data points can be used to produce more representative synthesised data to improve the performance of the system.

The experimentation has been followed up by the practical implementation of the KNN and SVM classifiers on a water meter developed in an ambient environment and trained using this data synthesis technique. The results produced, an example of which is shown in Figure 5.10, show that the live water meter produces comparable performance, using only 2 labelled data point examples per fixture event. Electrical disaggregation at the meter point has been briefly introduced but, the aim here is to address the suitability of such a technique to augment water meter disaggregation for data generation of activity recognition data, in the context of assisted living. The next chapter will describe the process of activity recognition carried out in an ambient environment developed to analyse the data produced from water meter and physically simulated electricity meter disaggregation systems.

Chapter 6 HAR using sensor Fusion

In chapter 4 two publicly available data sets (Crandall & Cook, 2008) and (Tapia, et al., 2004) were analysed to determine the contribution made by particular sensors to the classification of the activities that were monitored. The chapter described the implementation of six classification algorithms, chosen because of inherent and observable feature ranking, and presented the performance results of these algorithms using the full range of sensors and then after feature selection using a sensor subset obtained from ranking the features. The results confirmed that it is possible to reduce the number of sensors with no reduction in classifier performance and that a sensor subset containing data from electrical appliances, water usage sensors and motion sensors may provide enough data to recognise the activities of the occupant.

In chapter 5 methods of water and electricity disaggregation at the meter points were discussed and, to prove the viability of the concept, a method of water meter disaggregation was implemented using ANN, SVM and KNN classifiers. Two algorithms were presented that enabled the synthesis of training data from very few real world data point examples, thus demonstrating a significant reduction in implementation practicality and training times of such disaggregation systems. The experimentation and analysis of the water meter disaggregation system was followed up by the practical implementation of the KNN and SVM classifiers on a water meter developed in an ambient environment and trained using this data synthesis technique.

This chapter describes the implementation and analysis of an activity recognition system within that occupied ambient environment. Although the environment is a fully developed ambient environment, where the sensor implementation is ubiquitous, the experimentation carried out in this chapter is the culmination of the thesis aims and objectives and as such only a sub sensor set of electrical appliances, water usage (taken from water disaggregation) and PIR sensors are analysed. The activities of interest in this chapter are chosen by evaluation of the activities of the occupant, in situ rather than those described in chapter 2, the reasons for this are discussed herein. The full process of data collection, pre-processing and storage is described and an activity recognition data set that was collected over a period of 7 weeks is presented. For continuity the

same six classification algorithms are evaluated on the data set and the results analysed, before extending the feature set with the addition of a temporal feature representing the “Hour of Day” (Crandall & Cook, 2008).

The issues of imbalanced datasets have been discussed throughout this thesis and indeed the datasets that have been analysed to this point have been chosen because of the diversity between balanced and unbalanced activity distribution. It has been mentioned that real world activity recognition datasets are inherently unbalanced across the activity distribution and this is once again highlighted in the data that was collected and presented here. As this data imbalance has been a recurring theme it seems appropriate that a method of balancing the dataset using Synthetic Minority Over-sampling (SMOTE) is presented and implemented on the dataset before re-evaluation using the same classifiers and concluding with a discussion of the results.

6.1 Activity recognition in an Ambient Environment

A domestic ambient environment has been constructed by the researcher and has been specifically developed for activity recognition research. The core infrastructure of the sensor network has been developed with KNX technology (Richardson, 2015) in a wired topology that incorporates the complete lighting and heating system with power control of select devices such as water pumps and power sockets. The system has an integrated KNX smart electricity meter that provides 5 second interval measurements of aggregate Voltage, Current and Active power.

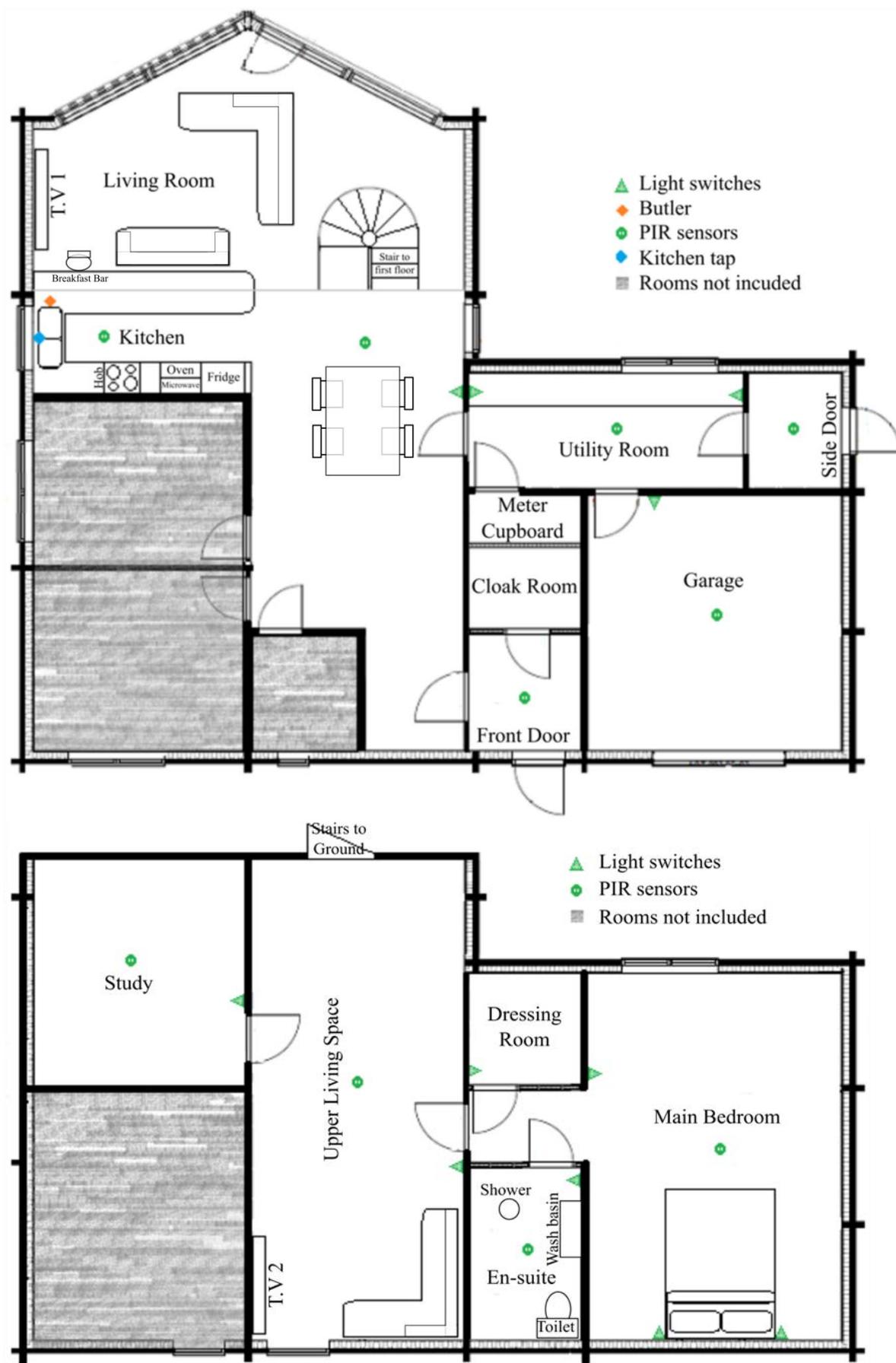
To facilitate ad hoc expansion and wireless integration the system incorporates a bridge into a wireless z-wave sensor network (Gomez & Paradells, 2010) which includes reed sensors on cupboard doors located in the kitchen and utility rooms, as well as device level power meters allowing fine grained metering of devices of choice. The network integrates the Sonos multi-room music system (Ubsdell, 2012), and this enables targeted messaging within the environment. Specific inhabitant control of the facility through several user interfaces is enabled by a Gira home server that includes both Android and Apple integration through applications on mobile phones and tablets. The domestic alarm system interfaces to the KNX network and so all alarm

alerts and messages as well as occupant interaction can be controlled and monitored through the alarm system, if required.

Configuration and development of the environment is facilitated at a high level by the use of vendor agnostic automation software, openHAB (Smirek, et al., 2014) running on an Intel Next Unit of Computing (NUC) small form factor computer. The openHAB software enables bridging between a z-wave network and, a smart watch, the Sonos system, Node-Red (Blackstock & Lea, 2014) using the Message Queuing Telemetry Transport (MQTT) protocol) as well as to external data sources generating a range of data from smart TV's to weather. Configuration and development at an embedded level, enabling the development and integration of custom built sensors, is facilitated by the serial to KNX embedded development board (SIM-KNX). Using SIM-KNX, a custom logging application has been developed and deployed to capture and timestamp all KNX messaging signals and these are saved to a database on the NUC. SIM-KNX has also been used to interface a purpose built smart water-meter, introduced in chapter 5, to the KNX system which receives PIR sensor data from the KNX network, and can also send water meter data onto the network. The smart water meter has been developed using a Raspberry Pi (Pi, 2012), Arduino (Russell, 2010) and the SIM-KNX development board.

The environment facilitates ambient monitoring of light switch actuation, temperature, light level, occupancy, water usage, leak detection, electricity usage (both coarse and fine grained) as well as control of device switching, water pumps, heating and cooling as well as the sound system. In total, the monitoring and control of over 200 data points have so far been configured and are being logged by the environment. Figure 6.1 shows the ground and First floor plan view of the ambient environment with the sensors that are relevant to this study highlighted and a key provided for clarity. Furniture is included in the plans only to provide context to the activities that were monitored, for example a "Breakfast Bar" is labelled as this is where the "Breakfast" activity took place, in most cases, and so the "Kitchen(PIR)" sensor is a dominant feature in the data point for this activity.

Figure 6.1: Ambient environment floor plans showing only those sensors used in the study.



Although over 200 data points are available for monitoring, the focus of this thesis is single occupancy environments with adapted smart meters and PIR sensors. With this in mind the environment was studied for a short period of time (pre-monitoring), to identify the predominant activity data points and hence the electrical appliance, water utilities, and PIR sensors used by the occupant. Also during this pre-monitoring period the single occupant, that is the subject of this study, did not use the rooms which are shown shaded in Figure 6.1 and so these were discluded from the monitoring process. The criterion for sensor inclusion was such as to match the thesis hypothesis and was guided by the work presented in chapter 4, as such only sensor events that represented movement (PIR sensor), Light switch activation (electrical disaggregation), appliance activation (electrical disaggregation) and water usage (water disaggregation) were considered. To enable filtration of the sensor data points of interest an MQTT broker was implemented and the openHAB software was configured to send MQTT messages for interesting sensor events only. This process creates some redundancy to mitigate the possibility of data loss and separates the raw data logging from the logging of the reduced sensor dataset. The floor plans in Figure 6.1 show the light switches of interest, the kitchen appliances of interest, the PIR sensors of interest and the kitchen tap and Butler are also shown on the ground floor plan. It should be noted that the butler is an instant boiled water tap for use when making a hot beverage or cooking, in an ordinary environment a kettle would be used to provide such a data point, but here water disaggregation can be used rather than electrical disaggregation for this device. The water utilities that were monitored are shown in the en-suite and the kitchen of Figure 6.1, and as the occupant did not use the bath during the monitoring period, the bathroom has been shaded on the first floor plan to discluded it form this study.

The smart water meter was developed and trained on two data point extremes, for each water utility used. In the kitchen the training data was taken from the kitchen tap and the Butler, and from the en-suite the training data was taken from the Shower, Wash basin and Toilet. A full set of labelled training data was then synthesised using the technique described in chapter 5 and SVM and KNN classifiers were trained using the synthesised data. This process of calibration and training is as discussed in chapter 5 and the entire process took a total of 8 mins 37 seconds

confirming the viability of this data synthesis technique to mitigate the issues of limited labelled training data and the practicalities of commissioning such systems in the field.

As discussed in chapter 5, full research and development of a smart electricity meter, that performs electrical disaggregation at the appliance level, is beyond the scope of this thesis and is left as future development work. Guidance taken from the work of (Gupta, et al., 2010), who showed that disaggregation at the device level using electrical signal noise was possible and could detect appliances such as televisions, refrigerator door lights and individual room lights as well as higher power events such as an Electric Oven, has informed the choice of appliances fitted with device level meters. The electrical appliances fitted with these meters were selected as Oven, Hob, Microwave, Toaster, Iron, Vacuum cleaner, Hair dryer, T.V1 and T.V2 and so a process of electrical disaggregation was simulated by implementing a combination of device level power meters for select electrical appliances, using KNX signalling for light fixture detection and a wireless reed sensor fitted to the fridge door to simulate the fridge door light.

The occupant being monitored did not use the Vacuum cleaner or hair dryer during the monitoring period and so no data points exist for these devices. This fact raises an important question as to the choice of activities to monitor and although the decision in the field of assisted living, as discussed in chapter 2, is often to monitor the IADL's (Lawton & Brody, 1988), such activities are often not part of the occupants every day activity set. Here a slightly different view point is considered using a pre-monitoring period to determine those activities that were carried out in the normal daily life of the occupant, and using these activities as the activity set of interest. It is envisaged that in future work a period of unsupervised learning could take place to find patterns in the data that represent specific activities and for these activities to be the focus of any activity recognition system. Using this method could result in a more bespoke and complete data flow of unbroken activity sequences, where the monitored activities are tailored to the occupant of the environment.

6.2 The Reduced Sensor set

The occupant was monitored for a period of 7 weeks and the sensor counts for the PIR, smart water meter and electrical sensors of interest are shown in the graphs of Figure 6.2 and Figure 6.3:

Figure 6.2: PIR and ground truth, water meter sensor counts for 7 week monitoring period

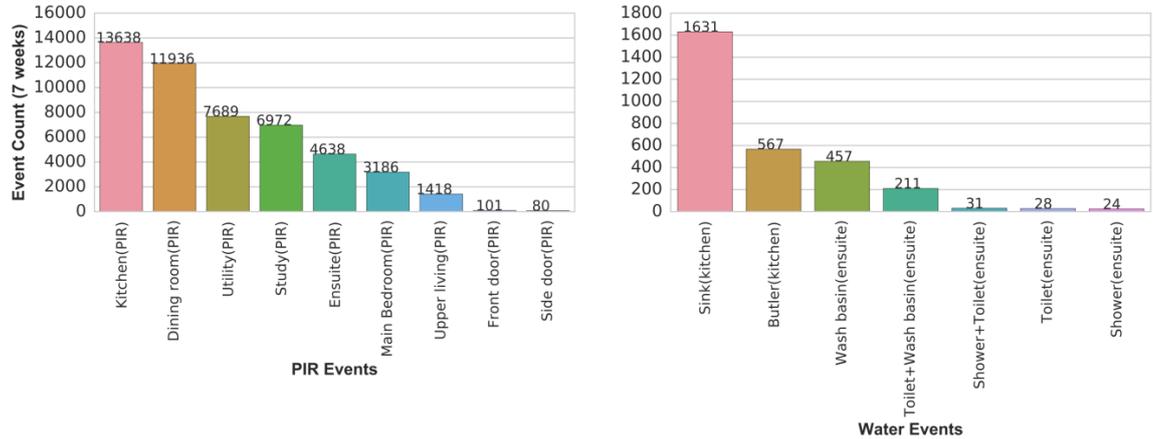
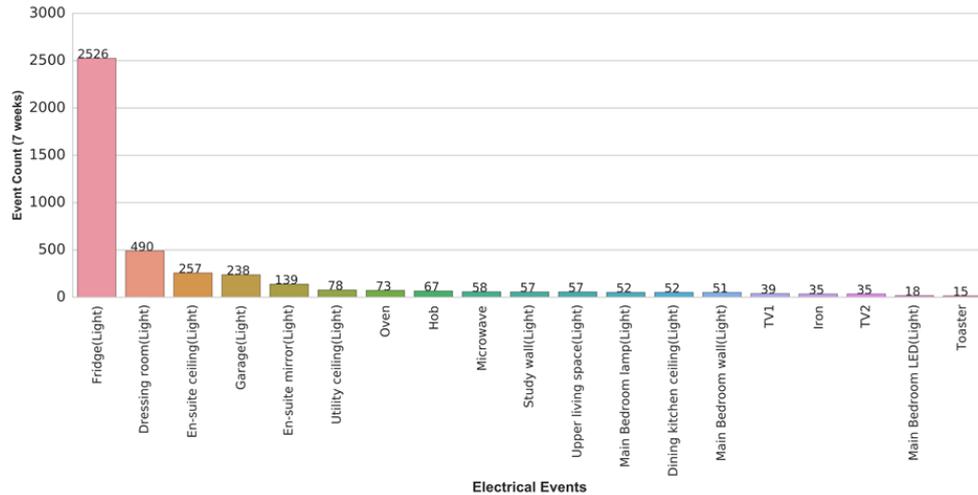


Figure 6.3: Electrical sensor counts for 7 week monitoring period



From the 35 sensors that were monitored there were a total of 56,944 sensor events recorded over the 7 week period. It is interesting to note that the most frequent PIR sensor events come from the kitchen and dining room areas identifying these as activity hotspots. Likewise the kitchen sink dominates the sensor count for disaggregated water events. The sensor counts for the relevant electrical appliances and lights are shown in Figure 6.3 noting the refrigerator light as the dominant sensor with 2,526 events; however it should be recognised that there is a habitual tendency to open and close the door of a refrigerator when using it, to maximise the efficiency of

the appliance. The machine learning framework, as discussed in chapter 3, was used to evaluate the supervised learning algorithms that were implemented in chapter 4, on the dataset that was collected here. The primary focus, guided by the results shown in chapter 4, is to evaluate the use of the reduced sensor set shown in Figure 6.2 and Figure 6.3 for activity recognition, using the chosen activities of interest. The following subsections describe the data collection and pre-processing that was carried out, with reference to the machine learning framework.

6.2.1 Data Collection

The data from the electricity and PIR sensors was redirected from its source using the OpenHAB Bridge and MQTT messaging protocol. Node-Red was then used to collect and log these MQTT messages into a series of files for later analysis. As the data from device level metering included power consumption data and each data point represented the instantaneous power consumption, transmitted on a change in data, Node-red was configured to collect this data point after a threshold value chosen to represent the appliance being switched on. For appliance activation (“on”) the threshold was used to identify a change from a low value to a high value, and for appliance de-activation (“off”) a high value to a low value (this was to mitigate issues with low level noise). The data between these thresholds was saved to an XML file, the form of which is shown in Figure 6.4, where the file name is used to determine the date when the data were recorded:

Figure 6.4: *Sample of an XML file showing a single Microwave event.*

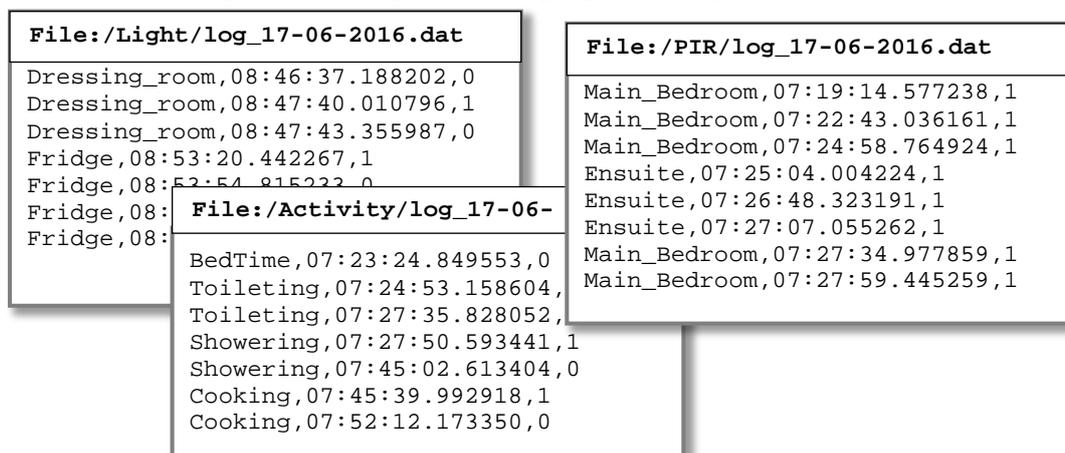
```
File: /Electricity/log_17_06_2016.dat

<?xml version="1.0"?>
<events>
<event>
  <node>Microwave</node>
  <start>08:55:14.823142</start>
  <data>0,1307.52,1297.92,1291.52,1278.08,2.2</data>
  <end>08:57:04.715428</end>
</event>
</events>
```

The PIR sensor, light sensor and activity ground truth labelling events only indicate timestamped (0) and (1) data, so in these cases Node-Red was configured to generate simple comma-separated value (CSV) files to represent this data, an example of which is shown in Figure 6.5. The figure

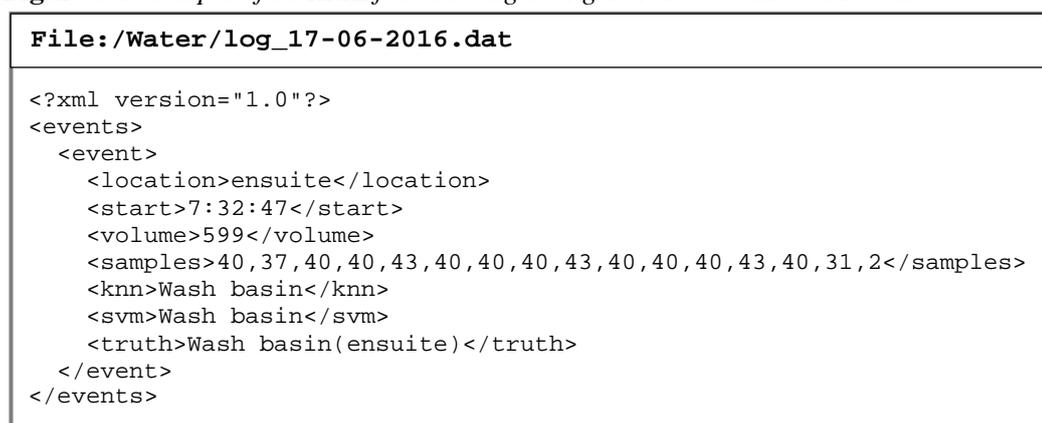
shows that for PIR sensors, only the activation (1) is recorded, this is justifiable as the PIR sensors were set to retrigger, approximately, every 30 seconds and the (0) value is an indication that there is no presence after this period, so this is not a signal generated directly by occupant presence.

Figure 6.5: Comma-separated value files showing Light, Activity and PIR data.



A useful side effect of using the PIR sensors in activity recognition is that a trace of movement throughout the environment from activity to activity can be maintained and used to evaluate the health and wellbeing of the occupant through movement analysis (Scanail, et al., 2006).

Figure 6.6: Sample of an XML file showing a single en-suite wash basin event.



The water meter data was also stored in the XML format, this is for future reference as the activation and subsequent deactivation is enough here to indicate interaction by the occupant with the utility of interest, a sample of water meter data is shown in Figure 6.6. Here the data recorded also includes the ground truth label as well as the labels predicted by the KNN and SVM classifiers. This has enabled the comparison of activity recognition performance using the ground

truth labels compared to labels produced by either classifier; however as the KNN classifier shows the most promising performance only labels produced by this classifier and ground truth are compared here. The water meter ground truth labels were annotated using purpose built online annotation software that enabled the analysis of all sensor events after the fact.

6.2.2 Data pre-processing and feature selection

As discussed in chapter 3, the Machine learning process used here is an offline process and so the files described in the previous section were used for data pre-processing. To ensure continuity with the activity recognition experimentation carried out in chapter 4, the features used for activity recognition here were sensor counts (“bag of sensors”) per activity (Cook & Krishnan, 2015). As all sensor and activity events during the monitoring period were time stamped the sensors were partitioned into those sensors that were active between the start and end of each activity thus creating the “bag of sensors”. Counts for each sensor were then extracted and so each feature was a count of the number of times the sensor was activated and/or deactivated during the duration of the activity. A sample of some of the resulting activity data points is shown in Table 6.1, and it should be noted that the “start_time” and “end_time” have not been included in this example for clarity and the ellipsis (...) indicate a break in the table that contains the columns representing the counts pertaining to the other sensors in the sensor set.

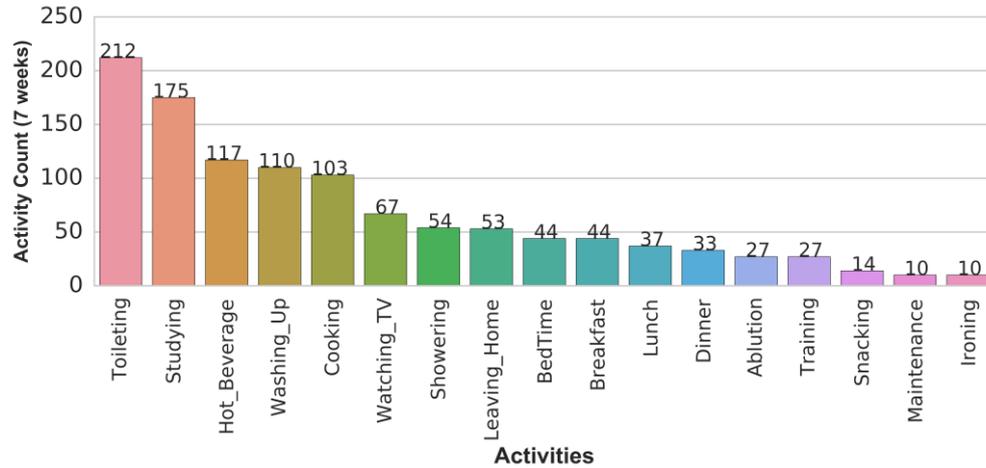
Table 6.1: *Sample of labelled activity data using sensor count as features.*

Activity	Hob	Butler	Ensuite(PIR)	Sink	...	Oven	Toilet + Wash basin	Utility(PIR)	Iron
Toileting	0	0	3	0	...	0	1	0	0
Cooking	1	0	0	6	...	0	0	4	0
Ironing	0	0	0	0	...	0	0	54	3
Washing_Up	0	0	0	6	...	0	0	0	0

6.3 The chosen activity set

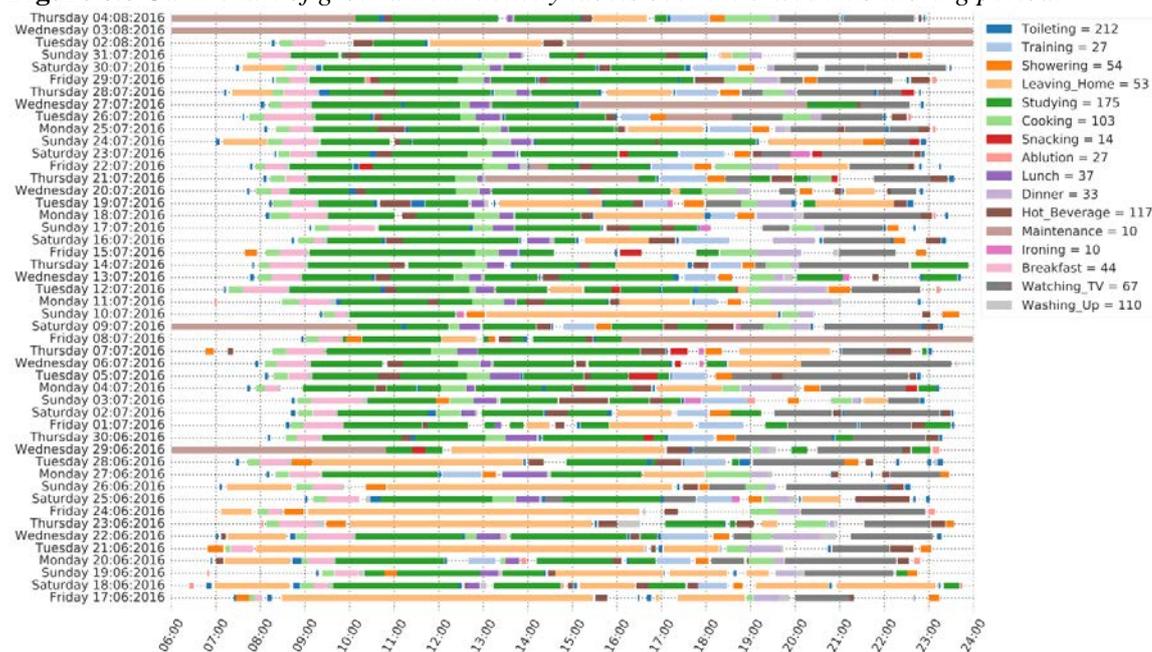
The activities were chosen during the pre-monitoring period so as to associate a relevant activity set with the occupant being monitored, the horizontal “Activities” axis of Figure 6.7 shows these activities:

Figure 6.7 Activities chosen and counts over the monitoring period



A smart phone and a smart watch application were configured to communicate with the ambient environment through the OpenHAB Bridge, which was in turn configured to send ground truth, time stamped labelling signals to indicate the start and end of an activity of interest over the 7 week monitoring period. The vertical axis of Figure 6.7 shows the activity counts generated for each activity over the full monitoring period. The graph indicates a total of 1,137 labelled activities were recorded over the 7 week period. From these the “Toileting” activity was the most frequent with 212 events followed by “Studying” (175) and having a “Beverage” (117), with “Ironing” (10) and “Maintenance” (10) the least frequent. The “Maintenance” activity represents periods where the data collection system was in failure or required maintenance, this activity was not included during activity learning and recognition. A Gantt chart was created using the ground truth activity labels and timestamps stored in the activity CSV files and is shown in Figure 6.5. The graph illustrates the patterns of activity that cover distinct time periods during any given day, this goes some way to validating the use of temporal features that were investigated in (Cook, et al., 2003) and (Crandall & Cook, 2008). For visibility the “Bedtime” activity, which represents the period that the occupant is in bed, is shown as areas of no activity early in the morning and late at night. Also for clarity the period between midnight and 6.00 am is not shown on the graph and was distinctive as a period of little activity that was dominated by the sensor events generated by the “Main Bedroom (PIR)” sensor.

Figure 6.8 Gantt chart of ground truth activity labels over the 7 week monitoring period.



The graphs of Figure 6.7 and Figure 6.8 illustrate a significant imbalance in the activity dataset, an issue discussed in chapters 2, 4 and 5 where it was highlighted as a major challenge in the field and will be investigated further here.

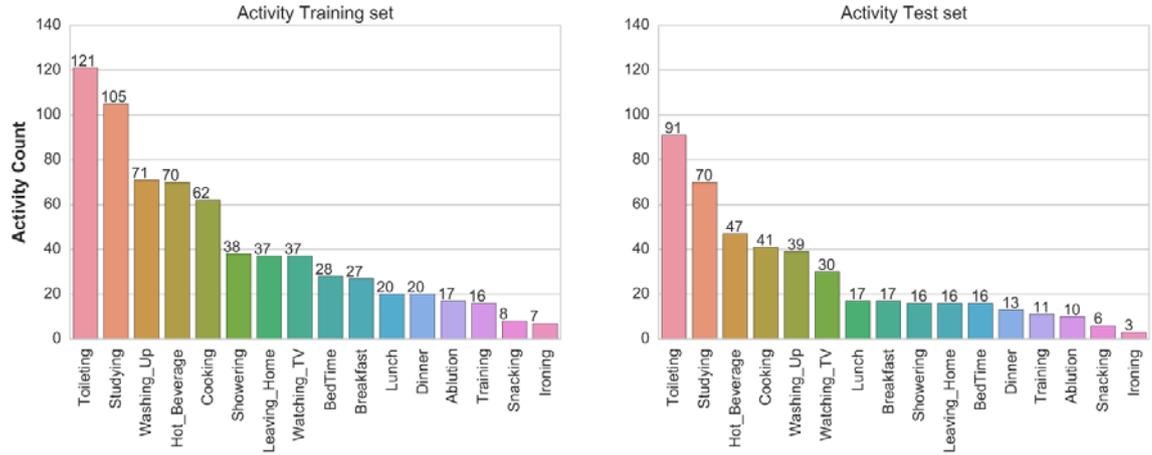
6.3.1 Activity segmentation

As an offline process of evaluation is carried out here, segmentation of the activities was achieved using a combination of user tagging of activity labels with a smart watch, and a mobile phone application. In addition, extensive data cleaning was carried out post collection, this was necessary as the environment was occupied by multiple residents and data that were not relevant to the user of interest, was filtered out post collection. For example if the Utility PIR sensor generated an event during a “Showering” activity of interest, the event was due to the other occupants using the Utility room while the monitored occupant was having a shower. It is acknowledged here that such extensive data sanitisation may not entirely relate to data that has been collected through an automatic online process of segmentation and so may result in better than average activity recognition results. Online segmentation and activity recognition is left as further research at this stage, where the lessons learned and results achieved here will form the basis of such research.

6.4 Evaluation of Machine Learning algorithms

The activity data was filtered to remove the “Maintenance” activity and so reducing the activity count to 1,127 activities that were then separated into training and test sets for algorithm evaluation. It was decided, in an effort to maintain proportionality of activity events between training and test sets, that the data should be split using periods of days. The justification for this is guided by the description of human behaviour provided in chapter 2, as purported by (Hull, 1943) human behaviour that is brought on by a specific need may be deterministic. The deterministic nature of human activity can be visualised, to some extent, in Figure 6.8 where daily patterns emerge for particular activities such as “Breakfast”, “Lunch”, “Dinner”, “Toileting” and “Bedtime”. These daily patterns reflect the basic biological and physiological human needs of food, excretion and sleep discussed by (Maslow, 1987). In light of this a 60/40 ratio split, of training data to test data, was used where the split was made using complete days rather than over activity data points. Over the 7 week period of monitoring a total of 47 days of data were recorded and so after the data split, the resulting data sets consisted of a short month (28 days) of training data and just under 3 weeks (19 days) of data were held out for testing. This reflects an initial 1 month period of non-intrusive, online data collection required to train the system. The training data therefore consisted of 684 activities over a 28 day period and the test data 443 activities over a 19 day period, interestingly these numbers are very close to the 676 to 451 split that would be achieved when splitting the activity set whilst ignoring the daily, habitual nature of the data. The resulting activity distribution over the training and test sets is shown in Figure 6.9. The graphs shows that the activity with the lowest count in the training set is “Ironing” with a count of 7, therefore stratified 7 fold cross-validation was used to determine the best model parameters using the F1-measure as the scoring method, and the parameter with the highest mean score chosen to parameterise the model.

Figure 6.9: Training and Test set split of activities.



The chosen models were then trained on the training data and tested on the held out test data, the results of which are shown in Table 6.2:

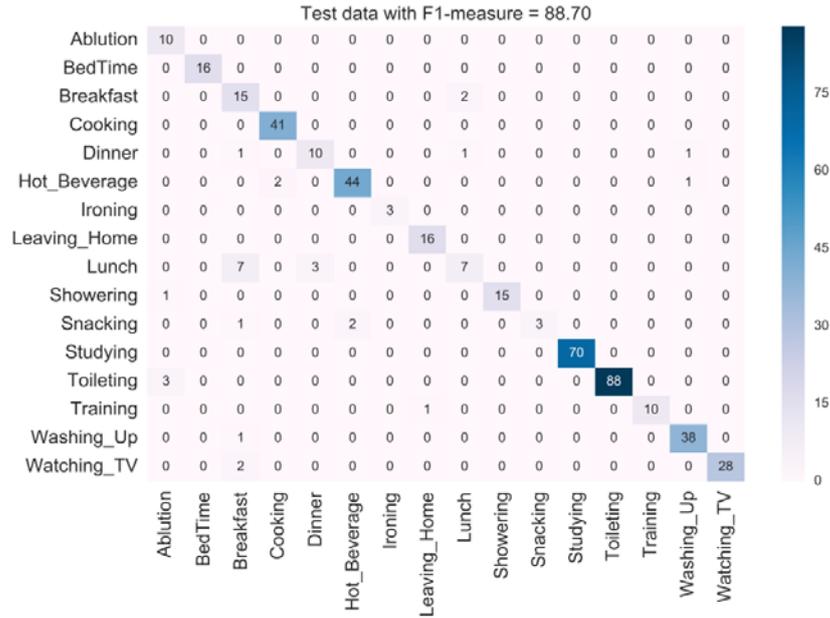
Table 6.2: Machine learning model evaluation using activity data collected over 7 weeks.

Machine Learning Model	Training set accuracy %	Test set accuracy %	Test set Precision %	Test set Recall %	Test set F1 score
NB (alpha=0.1)	89.91	88.71	85.76	82.36	82.04
SVM ((kernel=RBF, C=4.9)	97.81	82.62	70.93	66.47	66.67
SVM (kernel=linear, C=1.3)	96.78	93.45	90.88	88.88	88.70
LR (penalty = L1, C=4)	96.20	93.68	89.71	88.19	87.56
LR (penalty = L2, C=6.4)	95.91	93.91	91.22	88.56	88.25
DT (gini, depth=42)	99.56	92.10	84.62	85.86	84.35
DT (entropy, depth=8)	97.51	90.52	84.96	85.70	84.36
RF (gini, 77 learners, depth=42)	99.56	94.81	91.90	89.81	90.26
RF (entropy, 71 learners, depth=8)	97.81	93.91	90.88	88.04	88.19
Adaboost(gini, 91 learners, depth=42)	99.56	93.68	88.47	88.14	88.02
Adaboost(entropy, 65 learners, depth=8)	99.56	93.68	88.60	88.71	88.40

The results lie somewhere in between those presented in chapter 4 where the same algorithms were used to evaluate performance on the very clean, scripted dataset taken from the (Crandall & Cook, 2008) study and the much more natural (Tapia, et al., 2004) dataset.

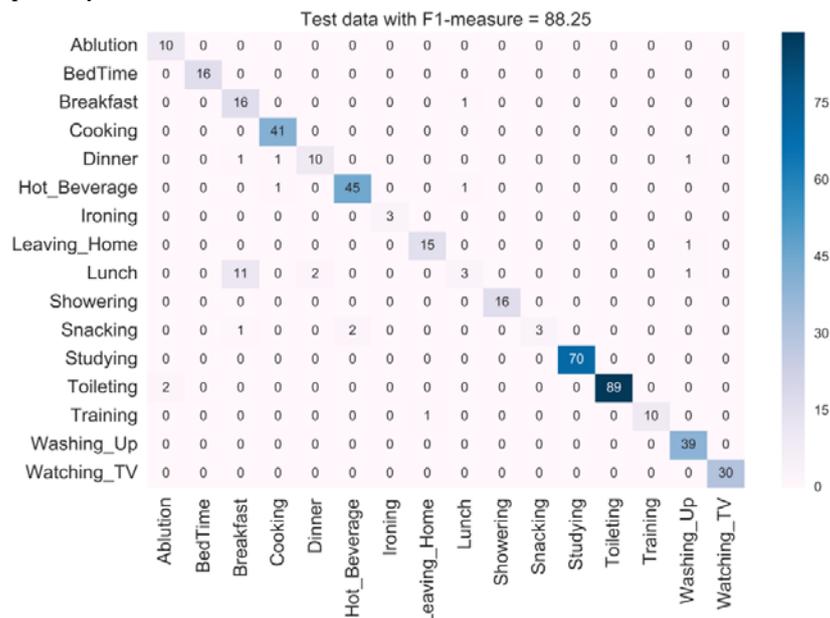
The Random Forest classifier using “gini” as the split criterion has outperformed the other classifiers but using “entropy” as the split criterion is only marginally worse, with Adaboost, Logistic Regression, and the SVM using a linear kernel also showing F1-measures in the region of 88. Further analysis was undertaken to determine if there are specific activities that contribute most to the misclassification cases. The confusion matrices of the best performing parameterised models for the SVM and Logistic Regression classifiers are shown in Figure 6.10 and Figure 6.11.

Figure 6.10: Test set confusion matrix for the SVM classifier using a linear kernel.



From inspection there is clearly confusion between the “Lunch” and “Breakfast” activities with 7 lunch activities being classified and labelled as “Breakfast”, using the SVM classifier and 11 such mistakes made by the LR classifier using the L2 penalty, there is also confusion between the “Lunch” and “Dinner” activities in both cases, and both show confusion between the “Snacking” and “Hot Beverage” activities which in hindsight could arguably be rationalised into a single activity.

Figure 6.11: Confusion matrix for Logistic Regression classifier using an L2 penalty.

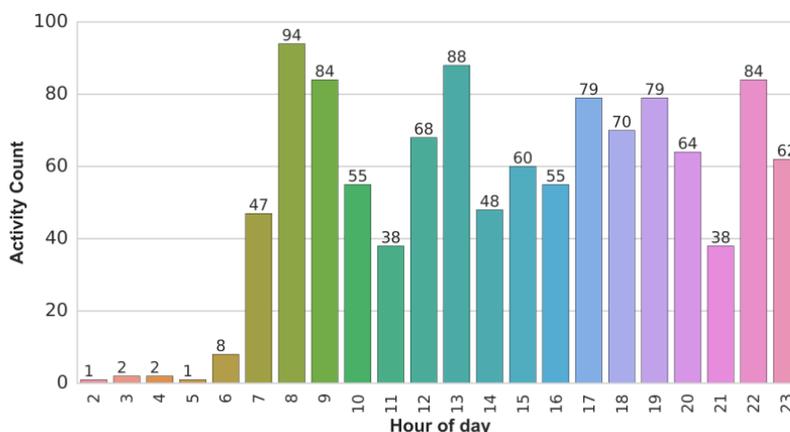


For all classifiers, with the exception of the Random Forest using gini and the linear SVM, the “Leaving Home” activity has at least once been wrongly classified as another activity e.g. the Logistic Regression classifier using the L2 penalty mistakes “Leaving Home” for the “Washing Up” activity, this is significant as such mistakes in a live, real time deployment could be much more safety critical than mistaking the “Toileting” activity with the “Ablution” activity. This anomaly will be revisited later in this chapter.

6.4.1 The addition of temporal features

As stated previously, the majority of confusion for each of the classifiers can be attributed to those activities that may be consider habitual, and so may have temporal properties that can be exploited, for example the activities of “Lunch” and “Dinner” may have very similar sensor signatures, but their very definition is determined by the time of day that they occur. With this in mind and using inspiration from the work of (Crandall & Cook, 2008), as discussed in chapter 2, where an “Hour of day” feature was successfully added to the feature set in an attempt to decrease classification false positives. The “Hour of day” feature was extracted from the timestamp of each activity and encoded as a categorical feature, the graph shown in Figure 6.14 shows the activity counts for each hour of the day for the entire data set and highlights distinctive rises in activity frequency between 7:00 and 10:00 i.e. breakfast time, 12:00 and 13:00 i.e. lunch time and 17:00 and 19:00 i.e. dinner time.

Figure 6.14: Activity counts at each hour of the day for the entire dataset.



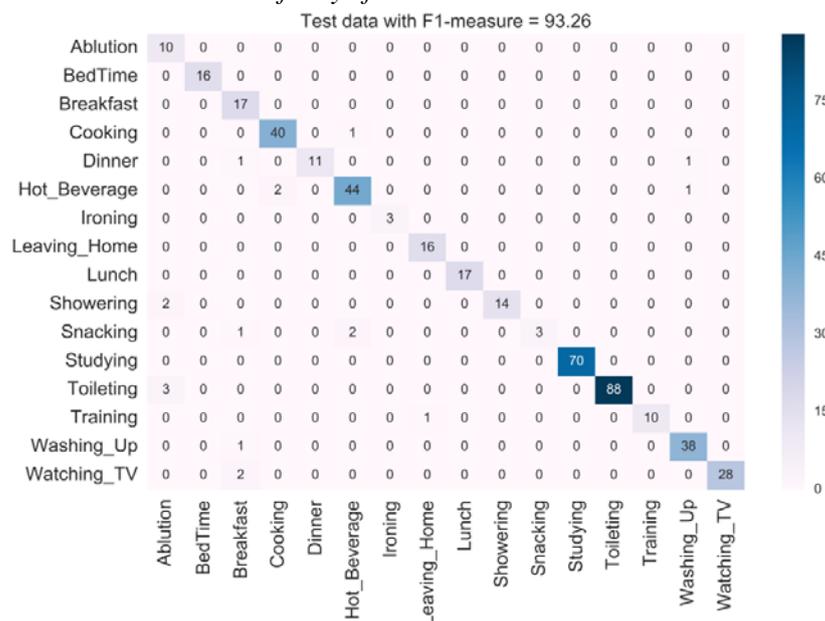
The feature was added to the training and test datasets and the process of cross-validation, training and testing, was repeated with the resulting performance metrics for each model shown in Table 6.3. The results show that the performance of all models has improved through the use of an “Hour of Day” temporal feature. For comparison Table 6.3 includes columns of F1-measures from models trained and tested without the “Hour of Day” (-HoD) and with (+HoD). The column representing the classification accuracy on the training set has been omitted in Table 6.3 for clarity, and it should be noted that the training set accuracy is usually only useful to gauge how well the model has fit the training data.

Table 6.3: Machine learning model evaluation with added “Hour of Day” feature.

Machine Learning Model	Test set accuracy %	Test set Precision %	Test set Recall %	Test set F1 -HoD	Test set F1 +HoD
NB (alpha=0.1)	93.00	91.03	88.91	82.04	88.98
SVM ((kernel=RBF, C=8.5)	85.56	78.93	73.64	66.67	74.44
SVM (kernel=linear, C=1.0)	99.71	95.94	95.12	93.23	93.26
LR (penalty = L1, C=3.7)	97.52	97.05	94.55	87.56	95.15
LR (penalty = L2, C=4.6)	97.52	97.05	94.55	88.25	95.15
DT (gini, depth=21)	95.26	91.72	91.30	84.35	90.38
DT (entropy, depth=84)	93.00	89.49	89.33	84.36	88.59
RF (gini, 91 learners, depth=21)	97.07	95.99	93.23	90.26	93.89
RF (entropy, 27 learners, depth=84)	95.71	94.68	91.84	88.19	92.41
Adaboost(gini, 71 learners, depth=21)	95.26	91.71	91.30	88.02	90.38
Adaboost(entropy, 11 learners, depth=84)	93.00	89.49	89.33	88.18	88.59

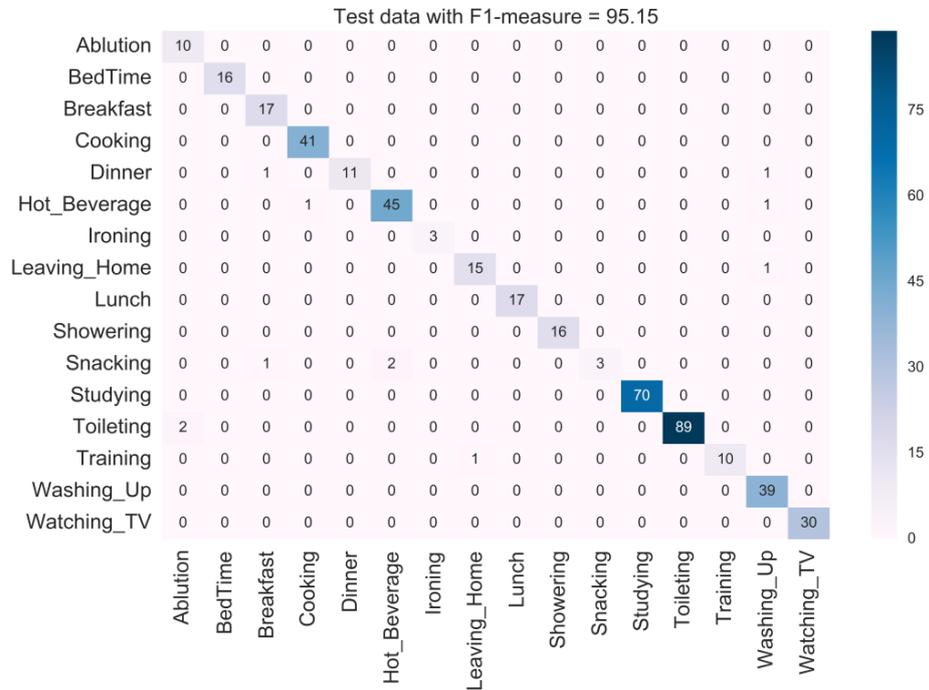
To determine how well the classifiers have differentiated between those activities with habitual temporal properties.

Figure 6.15: Confusion matrix for the SVM classifier using a linear kernel and an "Hour of Day" feature.



The confusion matrices were examined indicating that the confusion between activities that have temporal definitions have been improved across all models.

Figure 6.16: Confusion matrix for Logistic Regression classifier using an L2 penalty and an "Hour of Day" feature.

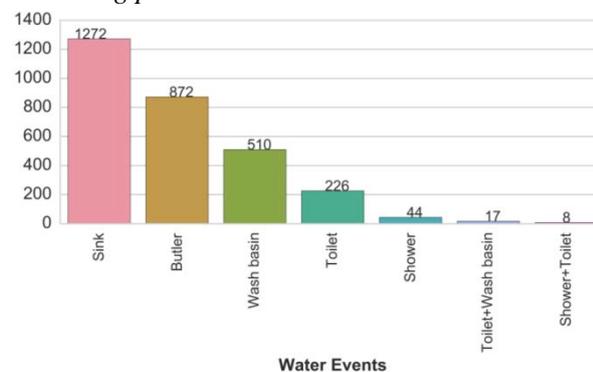


For comparison with previous results the confusion matrices for the linear SVM and Logistic Regression classifier using an L2 penalty are shown in Figure 6.15 and Figure 6.16. The matrices show that although the “Dinner” activity has been confused with “Breakfast” on one occasion the “Breakfast” and “Lunch” activities show perfect classification results. The issue of confusing the “Leaving Home” activity with the “Washing Up” activity is still a concern across all classifiers, with the exception of the Linear SVM. This highlights an important consideration when developing such systems for live application, in that, in some cases the best performing classifier evaluated using generic metrics such as F1-measure, may not be the best when there are critical activities that should always be recognised as they occur, and so the Linear SVM classifier may be a better option when choosing a classifier for an online HAR system where the “Leaving Home” activity should always be recognised.

6.4.2 Using disaggregated water data

So far in this chapter the classification algorithms have been evaluated using hand labelled ground truth water meter data i.e. all water data was taken for the field labelled “truth” from the XML water meter data files, a sample of which is shown in the file sample of Figure 6.6. To evaluate the use of data collected using the labels generated from the water meter classifiers, as described in chapter 5, the evaluation process was repeated using the labels classified by the meters KNN classifier. In practical terms the model evaluation procedure was repeated but instead of using the field labelled “truth” the field labelled “knn” from the XML water meter data files was used. Figure 6.17 shows that there is a significant difference in disaggregated counts from the ground truth counts that were shown in the right hand graph of Figure 6.2. In particular there is a decrease in “Sink” events and an increase in “Butler” events indicating that the KNN classifier is misclassifying and possibly confusing these events. Likewise a decrease in “Toilet + Wash basin” coupled with a large increase in “Toilet” events could also indicate some confusion between these particular events.

Figure 6.17: KNN water meter sensor counts for 7 week monitoring period.



The results of model selection using 7 fold cross-validation and testing is shown in Table 6.4 indicating extremely positive results using the proposed method of water meter disaggregation and training with only 2 data points per water utility. The results show that the Logistic regression classifier outperforms all other models with this dataset but most have seen a decrease in performance due to the loss in accuracy using classified water events over ground truth.

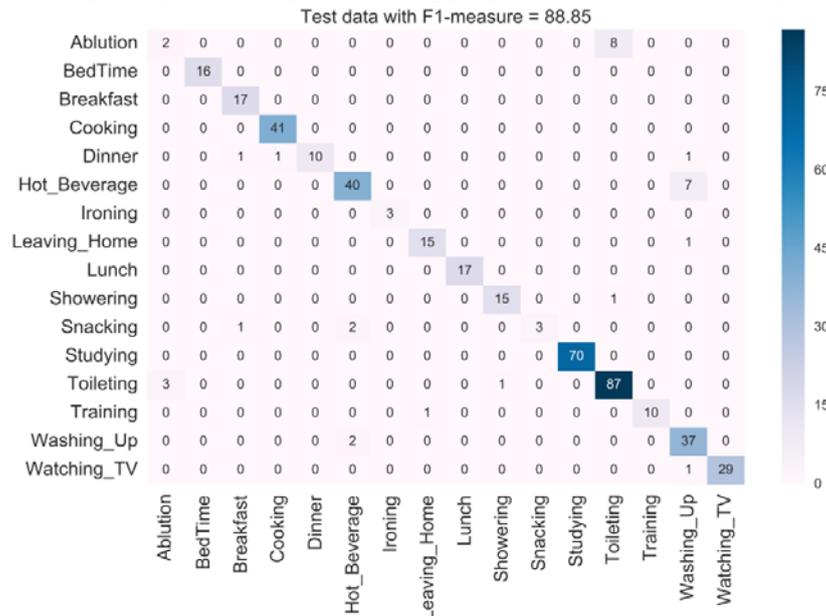
Table 6.4: Machine learning model evaluation using activity data with disaggregated water data using a KNN classifier.

Machine Learning Model	Training set accuracy %	Test set accuracy %	Test set Precision %	Test set Recall %	Test set F1 score
NB (alpha=0.1)	91.96	90.74	88.44	86.41	85.90
SVM ((kernel=RBF, C=5.5)	96.05	82.62	72.68	68.17	68.79
SVM (kernel=linear, C=1.0)	97.37	93.00	91.44	87.74	88.79
LR (penalty = L1, C=7.9)	97.67	93.00	92.18	87.35	88.85
LR (penalty = L2, C=5.8)	97.08	93.23	92.56	86.86	88.33
DT (gini, depth=67)	99.85	90.07	87.81	83.48	84.74
DT (entropy, depth=62)	99.85	89.84	86.90	82.14	83.68
RF (gini, 99 learners, depth=67)	99.85	92.55	91.25	84.97	86.61
RF (entropy, 51 learners, depth=62)	99.85	92.78	88.41	84.45	85.76
Adaboost(gini, 63 learners, depth=67)	99.85	90.52	87.49	83.45	84.70
Adaboost(entropy, 83 learners, depth=62)	99.85	90.29	87.65	85.18	85.38

The confusion matrix for the Logistic Regression classifier using the L1 penalty is shown in

Figure 6.18 and reveals the main areas of misclassification effected by the use of water disaggregation over ground truth for water utility events as the “Ablution” and “Hot Beverage” activities. As the “Ablution” activity is most often being labelled as a “Toileting” activity it can be implied that this confusion is possibly due to upstream misclassification at the water meter between “Wash basin” and “Toileting” events. Likewise the “Hot Beverage” activity is most often confused with the “Washing Up” activity and is likely due to confusion between the kitchen “Sink” and the “Butler”.

Figure 6.18: Confusion matrix for Logistic Regression classifier using an L1 penalty and using water disaggregation with a KNN classifier.



Both of these issues may be solved by incorporating more example data points when generating the synthesised training data for water meter disaggregation, so these results may be thought of as worst case with bare minimum of training at the water meter.

6.5 Balancing training data with synthesis

Although the classification algorithms chosen have been shown to perform reasonably well on the data set, the issue of misclassification of the “Leaving Home” activity remains a problem. As the dataset is inherently unbalanced, due to the proportionate nature of human activity, there is a possibility that data synthesis may help in producing models that are less bias to the high quantity of activities such as “Toileting”, “Hot Beverage” and “Washing Up”. In that the “Washing Up” activity is confused with the “Leaving Home” activity in the majority of the classifiers there is a chance that this could be due to model training bias and so this section describes the principles of and presents the results of applying Synthetic Minority Over-sampling (SMOTE) (Chawla, et al., 2002) in an attempt to balance the distribution of activities over the training data.

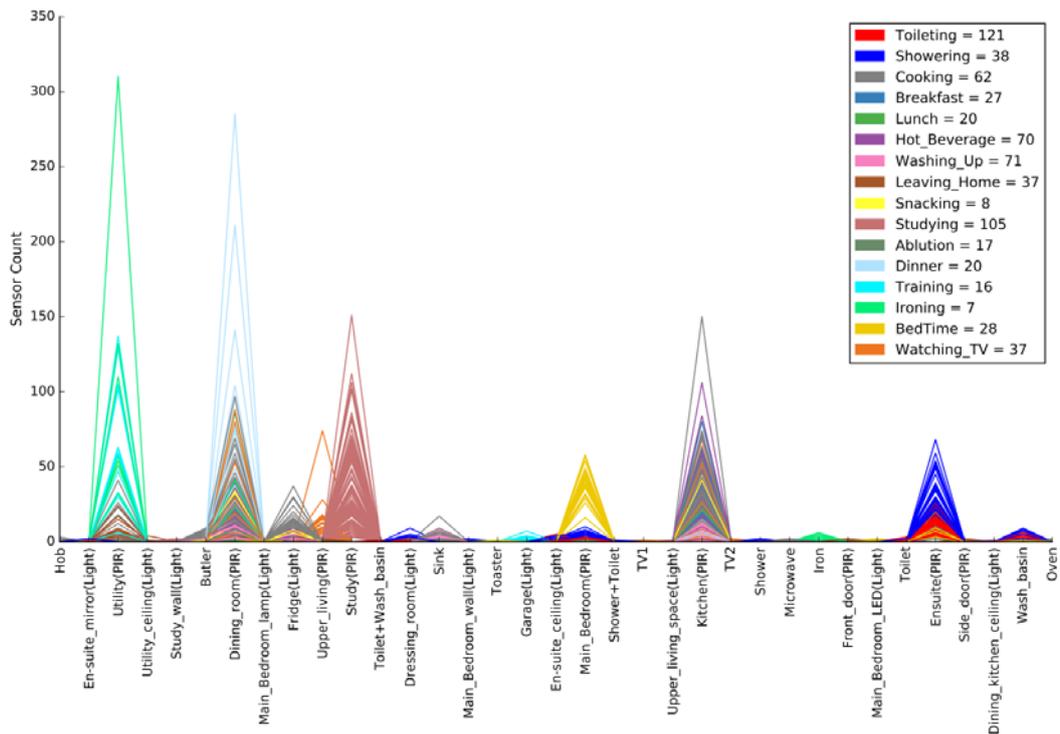
6.5.1 Synthetic Minority Over-sampling (SMOTE)

The principles behind SMOTE are comparable to the principles used to produce algorithms 1 and 2 in chapter 5 and were also inspired by work done in handwritten character recognition by (Ha & Bunke, 1997) and (Varga & Bunke, 2003). The notable difference in the development of SMOTE is the move from the data space, as described in the method implemented in chapter 5 as well as in (Ha & Bunke, 1997) and (Varga & Bunke, 2003), to the feature space. In SMOTE the new data points for the minority classes are synthesised using the KNN algorithm, where the nearest neighbors of each minority data point are used to synthesise new data points. Neighbors are randomly chosen and the difference between the feature vector of the minority data point, used for synthesis, and the chosen neighbour is multiplied by a random number between 0 and 1, and then added to the minority data point feature vector to produce a new data point. As stated in (Chawla, et al., 2002) the effective of this approach to data syntheses, or oversampling, is to force the decision region of the minority class to become more general, which effectively balances the bias/variance trade-off associated with the models produced.

6.5.2 Model evaluation using data from SMOTE

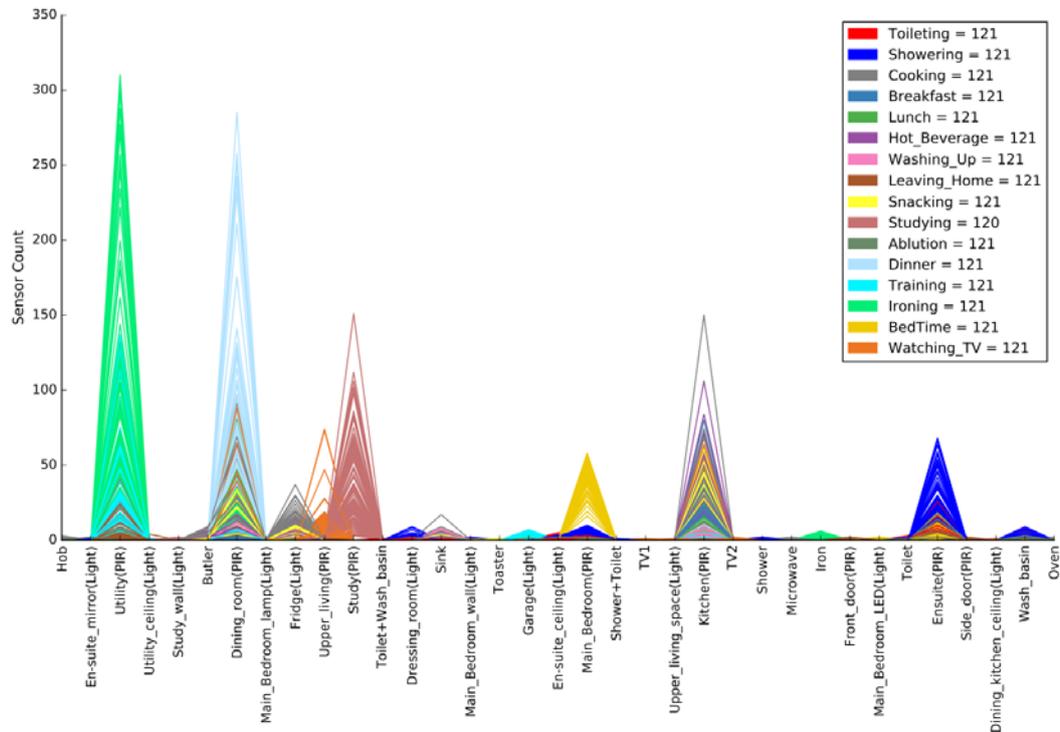
To evaluate whether data synthesis would improve the performance of the classifiers and in an effort to mitigate the issue associated with minority activities all algorithms were re-evaluated using data synthesis on the training set. The data was again split into training and test data using a 60/40 split both for the reasons described in the previous sections and for continuity of comparison of classifier performance. To compare the data before and after synthesis using the SMOTE technique, parallel coordinate plots are used to illustrate traces on a line graph that represent each activity along with the sensor count. The count is plotted on the vertical axis and sensor name on the horizontal axis. Such a plot is a general illustration of the most active sensors for a particular activity; Figure 6.19 shows the parallel coordinate plot for the training data before SMOTE is applied. Note that the “Time of Day” feature has been removed from the plot for aesthetic purposes only; the actual data set maintains the temporal features. The plot clearly shows that certain sensors are heavily involved in particular activities, for example the “Studying” activity is dominated by the “Study (PIR)” sensor, and a more subtle observation is the involvement of the “Iron” sensor in the “Ironing” activity.

Figure 6.19: Parallel coordinate plot showing sensor counts for each activity in the training set.



To implement SMOTE for synthesis of training data points to balance a data set, the method used here was to match the count of all activities with that of the majority class, and so balance the distribution of activities across the training set. Specifically for this application each activity was over sampled to a maximum of 121 activity counts to match that of the “Toileting” activity, it follows that no data synthesis was carried out for the “Toileting” activity. The data for each activity in the training dataset was isolated from the other activities and SMOTE used to synthesised data for that activity up to the majority class count; the separate data sets were then joined to the “Toileting” data to produce a new balanced training dataset. The parallel coordinate plot after data synthesis using SMOTE is shown in Figure 6.20. The plot shows that the synthesised data hasn’t changed the overall shape of the plot, but a distinct “colouring in” has changed the colour depth of the peaks in the graph. This indicates that the new data points have effectively created a more general version of each class by producing data points between neighbors of the KNN classification space, used during SMOTE synthesis.

Figure 6.20: Parallel coordinate plot showing sensor counts for each activity after SMOTE data synthesis of training data.



To investigate the effect of this synthesised balancing of training data on classifier performance, all algorithms used in the previous sections were evaluated using 10 fold cross-validation; this is

made possible as there is now no restriction to 7 folds caused by limited activity label distribution over the training data.

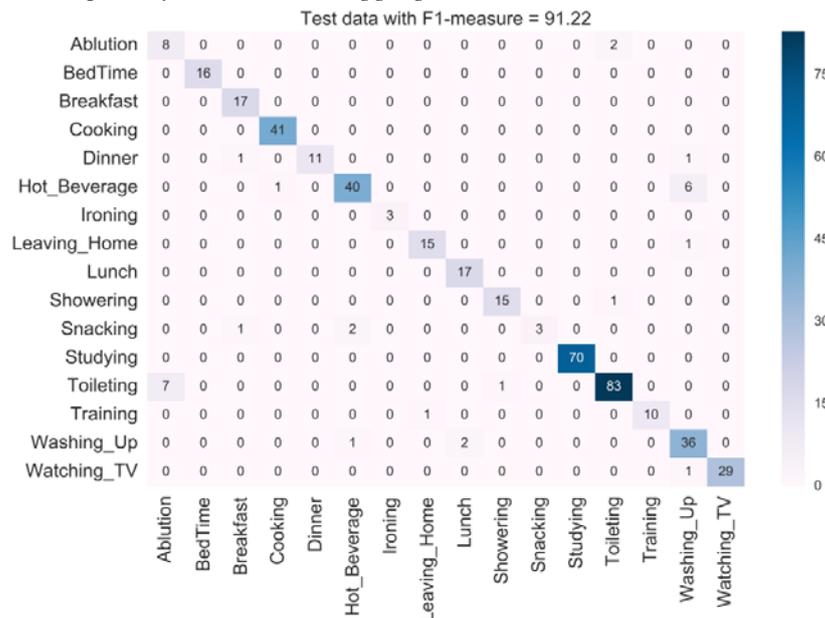
Table 6.5: Machine learning model evaluation using training data synthesised using the SMOTE technique.

Machine Learning Model	Training set accuracy %	Test set accuracy %	Test set Precision %	Test set Recall %	Test set F1 score
NB (alpha=)	93.70	89.62	84.30	86.91	84.91
SVM ((kernel=RBF, C=8.8)	99.07	85.32	80.42	78.49	77.80
SVM (kernel=linear, C=1.9)	99.28	92.10	91.69	89.30	89.47
LR (penalty = L1, C=6.7)	98.67	93.00	92.75	89.32	90.35
LR (penalty = L2, C=4.9)	98.29	93.45	92.93	91.14	91.22
DT (gini, depth=86)	99.95	92.10	89.21	87.66	87.96
DT (entropy, depth=64)	99.95	88.49	83.66	81.49	82.14
RF (gini, 70 learners, depth= 86)	99.95	93.68	94.18	87.02	89.20
RF (entropy, 45 learners, depth=64)	99.95	92.33	91.75	85.49	87.18
Adaboost(gini, 98 learners, depth=86)	99.95	92.78	90.07	86.39	87.75
Adaboost(entropy, 96 learners, depth=64)	99.95	90.07	86.71	83.21	84.54

The models were then trained using the best parameters for each and tested using the held out test set, the results of which are presented in Table 6.5 and show that the majority of the classifiers exhibit an increase in performance, with the exception of the NB classifier, the Decision Tree classifier using entropy, and the Adaboost classifier using entropy.

The best performing classifier was the Logistic Regression classifier using an L2 penalty and so the confusion matrix for this classifier is shown in Figure 6.21.

Figure 6.21: Confusion matrix for Logistic Regression classifier using an L2 penalty and water disaggregation and SMOTE.

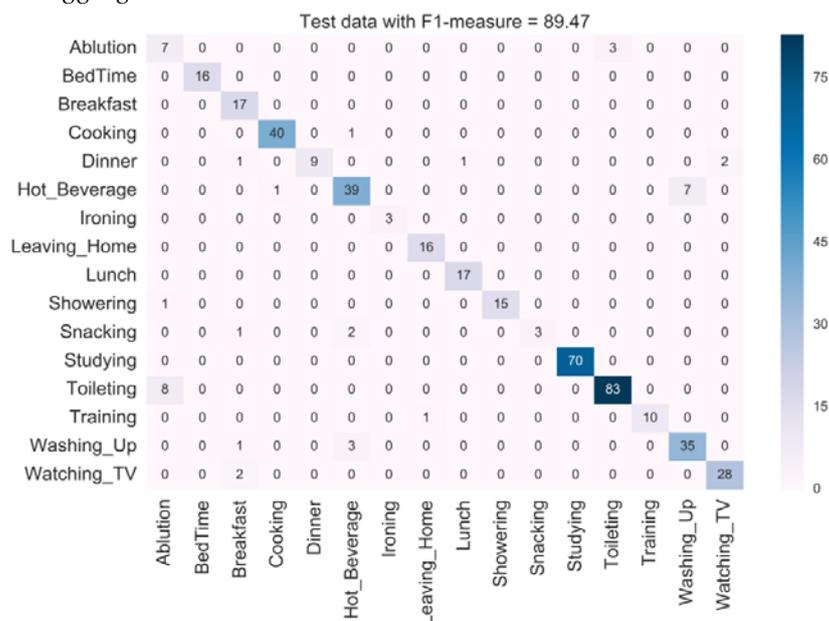


The matrix shows that although the performance has improved overall by balancing the training data through data synthesis using SMOTE, the majority of the confusion still comes from

activities that involve water utility fixtures, furthermore the “Leaving Home” activity is still miss-classified as “Washing Up” in one instance. To mitigate the issues regarding misclassification of critical activities, it may be necessary to select a model that performs marginally less well overall, but very well on the critical activity of interest.

In this case, although the linear SVM classifier performed marginally worse overall with an F1-measure of 89.47 and a classification accuracy of 92.10%, it may be a better option when the critical activity of “Leaving Home” is considered. The confusion matrix shown in Figure 6.22 indicates that the linear SVM performs perfectly when classifying the critical activity of “Leaving Home”.

Figure 6.22: Confusion matrix for a linear SVM classifier using water disaggregation and SMOTE.

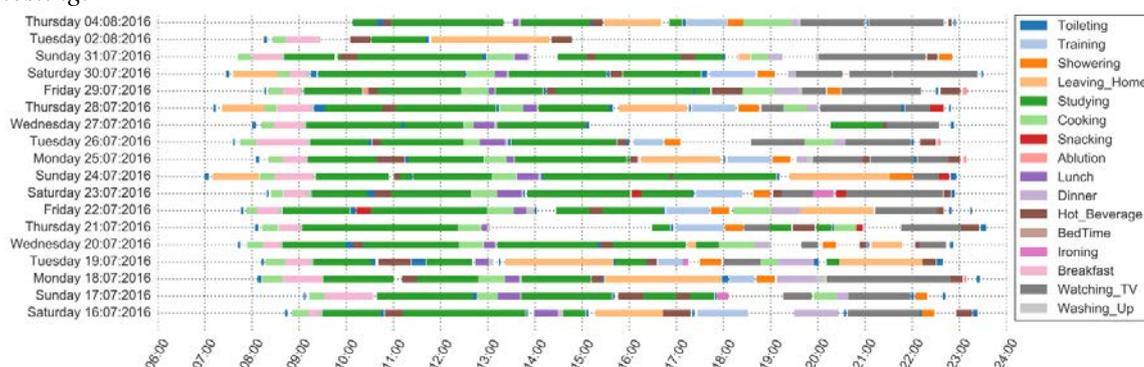


The matrix also shows that the majority of misclassifications involve water utility fixtures, in particular the activity of “Hot Beverage” is misclassified as a “Washing Up” activity 7 times, this could be due to, upstream, confusion between the “Butler” and “Sink” water fixtures and may be remedied with more data point examples when using data synthesis for disaggregation at the water meter.

The classifier that shows the best performance overall, using the “bag of sensor” counts and “Hour of day” as the feature set, and a training set balanced using over sampling of the minority

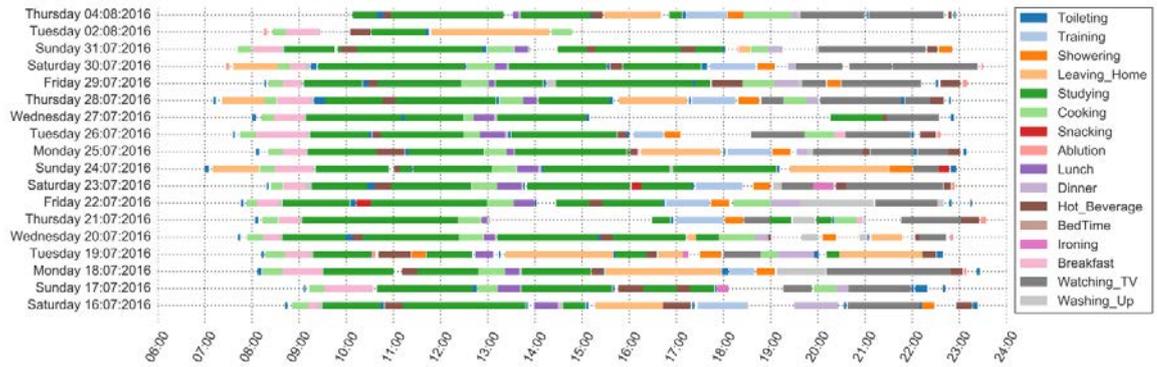
classes, with KNN disaggregated water meter data and considering each activity to be equally critical, is the Logistic Regression classifier using an L2 penalty. This classifier produces a classification accuracy of 93.45% and an F1-measure of 91.22, a visual representation for comparison of what this classification accuracy looks like in real terms is shown in Figure 6.23 and Figure 6.24. The ground truth data, used for testing the output of the classifiers, is presented in the Gantt chart of Figure 6.23, where each activity is colour coded and represented as a line in the graph where the horizontal axis is the “time of day” and the vertical axis is the “day of the week”.

Figure 6.23: Gantt chart showing 18 days of ground truth activity data used for algorithm testing.



This Gantt chart can be compared with the Gantt chart of the activities classified and labelled by the Logistic Regression classifier using an L2 penalty and is shown in Figure 6.24. The noteworthy difference in these graphs occurs on Friday 22:07:2016 at approximately 19:40 hours where the ground truth shown in Figure 6.23 indicates that the occupant leaves home and returns at around 21:15 hours, whereas in the classification by the LR classifier the occupant is labelled as having dinner and so is at home. If this activity is not considered critical then the two plots are remarkably similar and show that HAR in ambient environments using disaggregation at the meter points along with motion sensors in each room is a viable alternative to ubiquitous sensors or video surveillance techniques.

Figure 6.24: Gantt chart showing 18 days of activity data as predicted by the LR classifier using an L2 penalty, trained on data that was balanced using SMOTE, where water utility data was produced using KNN disaggregation.



6.6 Summary of Activity Recognition using sensor Fusion

This chapter has described the development of an ambient environment, shown in Figure 6.1 that has facilitated the experimentation and evaluation of a Human Activity recognition system that implements the fusion of utility meter disaggregation and movement sensor data for activity classification. Using supervised classification algorithms that match those used in chapter 4 a baseline evaluation was conducted by parameterising these algorithms using stratified cross validation and training the selected models using 28 days of activity data and testing the models using 19 days of activity data collected from a single occupant. The results, shown in Table 6.2, were analysed using precision, recall and F1-measures to determine the best performing classifiers and confusion matrices, shown in Figure 6.10, were examined to determine the activities that proved most troublesome to classify. From inspection it was found that some of those activities that were misclassified were habitual in nature and may benefit from a temporal feature to aid classification. The “Hour of Day” feature was implemented and the evaluation repeated with the results, shown in Table 6.3, demonstrating a marked improvement in the F1-measure for all models.

Prior to data collection, over the 7 week period, the water meter was trained using only 2 data points per water fixture, representing the extremes of water usage for that fixture with both KNN and SVM classifiers. The results demonstrated in Table 6.2 and Table 6.3 were recorded using the annotated ground truth labels for water usage data, and so the process was repeated using water data that was classified using the KNN classifier to determine fixture usage, as described in

chapter 5. The results using water disaggregation at the meter point were presented in Table 6.4, where a slight decrease in performance was registered due to the loss of performance in water meter classification, however as a worst case lower limit the results are promising.

Analysis of confusion matrices also highlighted the issue of critical activities such as “Leaving Home” and how some of the best performing classifiers fail to accurately classify this activity. Over sampling, in the form of SMOTE, was described as a method for synthesising training data in an attempt to balance the dataset and the evaluation process was repeated with a balanced, synthesised training set. On the analysis of the results, shown in Table 6.5, it was discovered that the use of data synthesis on the training data had improved the classifier performance but was not successful in aiding the classification of critical activities.

In summary this chapter has proven that the use of sensor data from water and electricity meter disaggregation, along with motion data from PIR sensors is enough to accurately recognise human activities with a classification accuracy of 93.45% and an F1-measure of 91.22 using the Logistic Regression classifier using an L2 penalty. When critical activities are considered it may be necessary to settle for classifiers that perform less well overall but classify these activities perfectly, for example it has been shown here that the “Leaving Home” activity can be perfectly classified by the Linear SVM classifier with a slightly lower overall accuracy of 92.10% and F1-measure of 89.47. It should however be noted that these are worst case performance metrics as the majority of misclassifications for each algorithm were during the classification of activities that used water fixtures events. These events were misclassified upstream at the point of water meter disaggregation and so with more example data points used for training data synthesis at the water meter, activity recognition can be improved.

Chapter 7 Conclusions and future work

The aim of this research has been to investigate a practically implementable system for non-intrusive monitoring of the elderly using current methods in the field of HAR. The novel system that has been developed incorporates a method of sensor fusion that implements disaggregation techniques, at both the water (Wonders, et al., 2016), and electricity meters (Gupta, et al., 2010), as well as movement data from PIR sensors to infer the activities of the inhabitants. The system has been implemented and evaluated using more computationally advanced versions of existing components of a standard home infrastructure. Specifically, the water meter has been repurposed to provide fixture level usage identification and individual device power meters have been used to simulate disaggregation at the electricity meter (Gupta, et al., 2010). Light switch activation and PIR sensor data are taken from a fully fitted ambient environment that has been purpose built for this research. The main challenges in activity recognition have been highlighted with a review of common approaches to human activity recognition, and the challenges of privacy and insufficient labelled training data are specifically addressed here, contributing significantly to the body of research.

To address the issues of privacy the idea of using low definition data is proposed and based on the principle that the higher the data definition the more invasive the monitoring process i.e. with more sensors the data reveals more detail on the subject as well as being more invasive to fit and maintain. It was proposed that by analysing the relevance of particular sensors to the classification problem fewer sensors may be used to achieve equivalent results. As such a Hybrid method for feature selection was developed that combined embedded feature selection contextualised by algorithm performance evaluation, extraction of ranked features followed by wrapper subset selection using backward elimination of the lowest ranking features. Classification algorithms, chosen because of ranked feature transparency, were used to implement the hybrid feature selection method. Two publicly available activity recognition datasets (Tapia, et al., 2004) and (Crandall & Cook, 2008) were evaluated which highlighted that the majority of sensor types in both cases were associated with movement, water or electricity

usage. Specifically using the (Tapia, et al., 2004) dataset it was shown that the number of sensors used in the dataset could be reduced from 72 to 43 with no reduction in classifier performance. This discovery formed a sound empirical basis for the research hypothesis and so was followed up with research and development of a system of water meter disaggregation.

A novel method of water meter disaggregation was demonstrated, in chapter 5, which compared the classification accuracy of ANN, SVM and KNN classifiers for predicting the fixture responsible for an event. Both labelled data collected over a two month period and synthesised data generated from only two extreme examples of labelled data per event class were used in this process. A significant contribution was made here, as for such a system to be practical, the implementation and training times must also be practical. It was demonstrated that by using synthesised data, automatically generated by algorithms developed from knowledge of the domain, that training times for water meter disaggregation can be reduced to minutes rather than hours or days (Srinivasan, et al., 2011). It was shown that the accuracy of 84.97% produced by the disaggregation system trained on synthesised data is comparable to other studies that do not use a synthesis process to produce training data (Chen, et al., 2005), (Fogarty, et al., 2006), (Froehlich, et al., 2009), (Srinivasan, et al., 2011), (Larson, et al., 2012), (Nguyen, et al., 2013a), and (Nguyen, et al., 2013b). The accuracy of 84.97% presented here is considered worst case, as only 2 labelled data point examples, that represent the extremes of water fixture events, are used to generate the synthesised training data and more example data points can be used to produce more representative synthesised data and so improve the performance of the system. The experimentation was followed up with the practical implementation of the water meter into an ambient environment and trained using this data synthesis technique.

An ambient environment, purpose built for HAR research, was developed and described in chapter 6. The environment was set up to implement the fusion of utility meter disaggregation and movement sensor data for activity classification and the same supervised classification algorithms used in chapter 4 were evaluated using 28 days of single occupant activity data and tested using 19 days of such data. The classification results using ground truth, hand labelled, water meter disaggregation (rather than classified disaggregation) data were presented and

analysed using precision, recall and F1-measures and confusion matrices examined to determine the activities that proved most difficult to classify. It was shown that the activities that were most often misclassified were habitual in nature and the addition of an “Hour of Day” (Crandall & Cook, 2008) feature was shown to provide marked improvement in the F1-measure for all models. An evaluation of activity classification using disaggregated water meter data that was classified using a KNN classifier followed. A slight decrease in performance was shown due to the loss of performance in water meter classification, but it is noted that these are with the worst case, minimum quantity of fixture data used for training at the water meter.

It was also demonstrated that by using over sampling, in the form of SMOTE, on the labelled activity training data, to balance out minority classes across the distribution of activities, the classification performance of all algorithms was improved. Analysis of confusion matrices highlighted the issue of critical activities such as “Leaving Home” and how some of the best performing classifiers fail to accurately classify this activity. It was demonstrated that when critical activities are considered it may be necessary to settle for classifiers that perform less well overall but classify these activities perfectly, for example it was shown that the “Leaving Home” activity can be perfectly classified by the Linear SVM classifier which had a slightly lower overall accuracy of 92.10% and F1-measure of 89.47, compared to the best classification accuracy of 93.45% and F1-measure of 91.22 shown by the Logistic Regression classifier using an L2 penalty.

In summary it has been demonstrated through research, development and evaluation that a system for non-intrusive monitoring within an ambient environment, occupied by a single resident, is achievable using repurposed versions of the standard domestic infrastructure. More specifically it has been shown that a minimum baseline accuracy of 93.45% and F1-measure of 91.22 can be achieved using disaggregation at the water and electricity meters combined with locality context provided by home security PIR sensors. Methods of speeding up the deployment and commissioning process have been proposed and evaluated and proven to be viable, further demonstrating the potential practical application of such methods.

7.1 Future work

Although the work presented in this thesis has provided a basis to address the challenges of privacy and insufficient labelled training data the remaining challenges of intra and inter-class variations in the data and multiple occupancy have not been addressed here. Also the use of individual appliance power meters combined with light switch activation data from a fully fitted ambient environment is a limitation of the research presented.

It is proposed that further research, development and evaluation work be carried out to determine the viability of repurposing the electricity meter with similar functionality to the water meter presented by building on the work published by (Gupta, et al., 2010) with the inclusion of classifier training using synthesised data, and so allowing a complete system evaluation of classification of disaggregated events at both the water and the electricity meters.

In terms of the challenges created by multiple occupancy environments a means of identifying the occupants within the environment is a proposal for future work. Building on the work published by (Azghandi, et al., 2015) and (Benmansour, et al., 2016) where data from RFID tags, augmented ambient environment data, to both identify and localise the residents. Future work will investigate the viability of using light detection and ranging (LIDAR) sensors for identification and localisation in place of the PIR sensor used in this research and RFID sensors used by (Azghandi, et al., 2015) and (Benmansour, et al., 2016).

References

- Abu-Mostafa, Y. S., Magdon-Ismail, M. & Lin, H.-T., 2012. *Learning from data*. s.l.:AMLBook Singapore.
- Ainsworth, B. E. et al., 2000. Compendium of physical activities: an update of activity codes and MET intensities. *Medicine and science in sports and exercise*, Volume 32, pp. S498--S504.
- Al-Bin-Ali, F., Boddupalli, P., Davies, N. & Friday, A., 2003. *Correlating sensors and activities in an intelligent environment: A logistic regression approach*. s.l., s.n., pp. 318-333.
- Alelyani, S., Tang, J. & Liu, H., 2013. Feature Selection for Clustering: A Review. *Data Clustering: Algorithms and Applications*, Volume 29.
- Alpaydin, E., 2014. *Introduction to Machine Learning*. s.l.:The MIT Press.
- Aminian, K. et al., 1999. Physical activity monitoring based on accelerometry: validation and comparison with video observation. *Medical Biological Engineering Computing*, 37(3), pp. 304-308.
- Anjum, A. & Ilyas, M., 2013. *Activity recognition using smartphone sensors*. s.l., s.n., pp. 914-919.
- Anon., 2007. *Oxford English Dictionary*. 3 ed. s.l.:Oxford University Press.
- Atallah, L., Lo, B., King, R. & Yang, G.-Z., 2011. Sensor Positioning for Activity Recognition Using Wearable Accelerometers. *Biomedical Circuits and Systems, IEEE Transactions on*, Aug, 5(4), pp. 320-329.
- Augusto, J., Nakashima, H. & Aghajan, H., 2010. Ambient Intelligence and Smart Environments: A State of the Art. In: H. Nakashima, H. Aghajan & J. Augusto, eds. *Handbook of Ambient Intelligence and Smart Environments*. s.l.:Springer US, pp. 3-31.
- Austin, D. et al., 2011. On the Disambiguation of Passively Measured In-home Gait Velocities from Multi-person Smart Homes. *J. Ambient Intell. Smart Environ.*, April, 3(2), pp. 165-174.

- Ayers, D. & Shah, M., 2001. Monitoring human behavior from video taken in an office environment. *Image and Vision Computing* , 19(12), pp. 833-846.
- Azghandi, M. V., Nikolaidis, I. & Stroulia, E., 2015. *Multi--Occupant Movement Tracking in Smart Home Environments*. s.l., s.n., pp. 319-324.
- Bae, I.-H., 2014. An ontology-based approach to ADL recognition in smart homes. *Future Generation Computer Systems*, Volume 33, pp. 32-41.
- Barber, D., 2006. *Machine Learning A Probabilistic Approach*.
- Barlow, H. B., 1989. Unsupervised learning. *Neural computation*, 1(3), pp. 295-311.
- Barngrover, C., Kastner, R. & Belongie, S., 2015. Semisynthetic Versus Real-World Sonar Training Data for the Classification of Mine-Like Objects. *Oceanic Engineering, IEEE Journal of*, Jan, 40(1), pp. 48-56.
- Bayat, A., Pomplun, M. & Tran, D. A., 2014. A Study on Human Activity Recognition Using Accelerometer Data from Smartphones. *Procedia Computer Science* , Volume 34, pp. 450-457.
- Ben-hur, A. & Weston, J., 2007. *A user's guide to support vector machines*.
- Benmansour, A., Bouchachia, A. & Feham, M., 2016. Multioccupant activity recognition in pervasive smart home environments. *ACM Computing Surveys (CSUR)*, 48(3), p. 34.
- Bharucha, A. et al., 2009. Intelligent assistive technology applications to dementia care: current capabilities, limitations, and future challenges.. *The American Journal of Geriatric Psychiatry: Official Journal of the American Association for Geriatric Psychiatry*, Volume 17, pp. 88-104.
- Bishop, C., 2006. *Pattern Recognition and Machine Learning*. s.l.:Springer.
- Blackstock, M. & Lea, R., 2014. *Toward a distributed data flow platform for the web of things (distributed node-red)*. s.l., s.n., pp. 34-39.
- Blum, A. L. & Langley, P., 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence* , 97(1-2), pp. 245-271.

- Bobick, A. F., 1997. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 352(1358), pp. 1257-1265.
- Boser, B. E., Guyon, I. M. & Vapnik, V. N., 1992. *A Training Algorithm for Optimal Margin Classifiers*. New York, NY, USA, ACM, pp. 144-152.
- Bose, R. & Helal, A., 2008. *Observing Walking Behavior of Humans Using Distributed Phenomenon Detection and Tracking Mechanisms*. s.l., s.n., pp. 405-408.
- Box, G. E. & Draper, N. R., 1987. *Empirical model-building and response surfaces*. s.l.:Wiley New York.
- Brand, M. & Kettner, V., 2000. Discovery and segmentation of activities in video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8), pp. 844-851.
- Breiman, L., 1984. *Classification and regression trees*. Belmont, Calif: Wadsworth International Group.
- Breiman, L., 1996. Bagging predictors. *Machine learning*, 24(2), pp. 123-140.
- Brooks, K., 2003. *The context quintet: narrative elements applied to context awareness*. s.l., Erlbaum Associates, Inc..
- Bulling, A., Blanke, U. & Schiele, B., 2014. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Comput. Surv.*, January, 46(3), pp. 33:1--33:33.
- Byvatov, E., Fechner, U., Sadowski, J. & Schneider, G., 2003. Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification. *Chemical Information and Computer Science*, 43(6), pp. 1882-1889.
- Caspersen, C. J., Powell, K. E. & Christenson, G. M., 1985. Physical activity, exercise, and physical fitness: definitions and distinctions for health-related research.. *Public health reports*, 100(2), p. 126.

- Chandrashekar, G. & Sahin, F., 2014. A survey on feature selection methods. *Computers and Electrical Engineering*, 40(1), pp. 16-28.
- Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, pp. 321-357.
- Chen, J. et al., 2005. Bathroom Activity Monitoring Based on Sound.. *Pervasive Computing, Lecture Notes in Computer Science*, Volume 3468, pp. 47-61.
- Chen, L. et al., 2012. Sensor-Based Activity Recognition. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Nov, 42(6), pp. 790-808.
- Chen, W.-H., Hsu, S.-H. & Shen, H.-P., 2005. Application of SVM and ANN for intrusion detection. *Computers & Operations Research*, Volume 32, pp. 2617-2634.
- Chen, Y. H. et al., 2009. *Preference model assisted activity recognition learning in a smart home environment*. s.l., s.n., pp. 4657-4662.
- Cook, D. J., Krishnan, N. C. & Rashidi, P., 2013. Activity discovery and activity recognition: A new partnership. *IEEE transactions on cybernetics*, 43(3), pp. 820-828.
- Cook, D. J. & Schmitter-Edgecombe, M., 2009. Assessing the quality of activities in a smart environment. *Methods of information in medicine*, 48(5), p. 480.
- Cook, D. & Krishnan, N., 2015. *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*. s.l.:Wiley-Blackwell.
- Cook, D. et al., 2009. *Collecting and disseminating smart home sensor data in the CASAS project*., s.l., s.n.
- Cook, D. et al., 2003. *MavHome: an agent-based smart home*. s.l., s.n., pp. 521-524.
- Cover, T. & Hart, P., 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, Volume 13, pp. 21-27.

- Cox, D. R., 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 215-242.
- Crandall, A. & Cook, D., 2008. *Attributing events to individuals in multi-inhabitant environments*. s.l., s.n., pp. 1-8.
- Crandall, A. & Cook, D., 2010. *Using a Hidden Markov Model for Resident Identification*. s.l., s.n., pp. 74-79.
- Crandall, A. S. & Cook, D. J., 2009. Coping with Multiple Residents in a Smart Environment. *J. Ambient Intell. Smart Environ.*, December, 1(4), pp. 323-334.
- Das, S., 2001. *Filters, wrappers and a boosting-based hybrid for feature selection*. s.l., s.n., pp. 74-81.
- Emiliani, P. L. & Stephanidis, C., 2005. Universal access to ambient intelligence environments: opportunities and challenges for people with disabilities. *IBM Systems Journal*, 44(3), pp. 605-619.
- Fang, H. et al., 2014. Human activity recognition based on feature selection in smart home using back-propagation algorithm. *ISA transactions*, 53(5), pp. 1629-1638.
- Feuz, K. D. & Cook, D. J., 2015. Transfer Learning Across Feature-Rich Heterogeneous Feature Spaces via Feature-Space Remapping (FSR). *ACM Trans. Intell. Syst. Technol.*, 6(1), pp. 3:1--3:27.
- Fishkin, K., Philipose, M. & Rea, A., 2005. *Hands-on RFID: wireless wearables for detecting use of objects*. s.l., s.n., pp. 38-41.
- Fogarty, J., Au, C. & Hudson, S., 2006. *Sensing from the Basement: A Feasibility Study of Unobtrusive and Low-Cost Home Activity Recognition*. s.l., s.n.
- Freund, Y. & Schapire, R. E., 1995. *A decision-theoretic generalization of on-line learning and an application to boosting*. s.l., s.n., pp. 23-37.

- Friedman, J. H., 1997. On Bias, Variance, 0/1---Loss, and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery*, 1(1), pp. 55-77.
- Friedman, N., Geiger, D. & Goldszmidt, M., 1997. Bayesian network classifiers. *Machine learning*, 29(2-3), pp. 131-163.
- Froehlich, J. E. et al., 2009. *HydroSense: Infrastructure-mediated Single-point Sensing of Whole-home Water Activity*. New York, NY, USA, ACM, pp. 235-244.
- Fuentes, D., Gonzalez-Abril, L., Angulo, C. & Ortega, J., 2012. Online motion recognition using an accelerometer in a mobile device. *Expert Systems with Applications*, 39(3), pp. 2461-2465.
- Gardner, M. & Dorling, S., 1998. Artificial Neural Networks (the Multilayer Perceptron) - A Review of Applications in the Atmospheric Sciences. *Atmospheric Environment*, Volume 32, pp. 2627-2636.
- Garg, V. et al., 2014. Privacy concerns in assisted living technologies. *annals of telecommunications - annales des télécommunications*, 69(1-2), pp. 75-88.
- Garg, V. et al., 2014. Privacy concerns in assisted living technologies. *annals of telecommunications - annales des télécommunications*, 69(1-2), pp. 75-88.
- Gomez, C. & Paradells, J., 2010. Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6), pp. 92-101.
- Gupta, S., Reynolds, M. & Patel, S., 2010. *ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the Home*. s.l., s.n., pp. 139-148.
- Guyon, I., Gunn, S., Nikravesh, M. & Zadeh, L. A., 2006. *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc..
- Hable, R. & Christmann, A., 2011. On Qualitative Robustness of Support Vector Machines. *Journal of Multivariate Analysis*, Volume 102, pp. 993-1007.

- Hart, G., 1992. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, Dec, 80(12), pp. 1870-1891.
- Ha, T. M. & Bunke, H., 1997. Off-line, handwritten numeral recognition by perturbation method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), pp. 535-539.
- Hayes, T., Pavel, M. & Kaye, J., 2008. An approach for deriving continuous health assessment indicators from in-home sensor data. In: *Assistive Technology Research Series*. s.l.:s.n., pp. 130-137.
- He, H. & Garcia, E. A., 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, Sept, 21(9), pp. 1263-1284.
- Helal, A. et al., 2012. *3D Modeling and Simulation of Human Activities in Smart Spaces*. s.l., s.n., pp. 112-119.
- Helal, A., Cook, D. J. & Schmalz, M., 2009. Smart home-based health platform for behavioral monitoring and alteration of diabetes patients. *Journal of diabetes science and technology*, 3(1), pp. 141-148.
- Helal, S. et al., 2011. *Persim - Simulator for Human Activities in Pervasive Spaces*. s.l., s.n., pp. 192-199.
- Helal, S. et al., 2005. The Gator Tech Smart House: a programmable pervasive space. *Computer*, March, 38(3), pp. 50-60.
- Help The Aged, 2009. *Re-shaping our society for older people*. s.l.:Age UK.
- Ho, T. K., 1995. *Random decision forests*. s.l., s.n., pp. 278-282.
- Ho, T. K., 1998. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8), pp. 832-844.
- Hsu, C.-W. & Lin, C.-J., 2002. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2), pp. 415-425.

- Hull, C. L., 1943. *Principles of behavior: an introduction to behavior theory..* s.l.:Appleton-Century.
- Hyafil, L. & Rivest, R. L., 1976. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1), pp. 15-17.
- Japkowicz, N. & others, 2000. *Learning from imbalanced data sets: a comparison of various strategies.* s.l., s.n., pp. 10-15.
- Jordan, A., 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, Volume 14, p. 841.
- Joyce, J., 2008. Bayes' Theorem. In: E. N. Zalta, ed. *The Stanford Encyclopedia of Philosophy*. Fall 2008 ed. s.l.:s.n.
- Juan Carlos Augusto, P. M., 2007. Ambient Intelligence: Concepts and Applications. *Computer Science and Information Systems*, Issue 7, pp. 1-28.
- Kabir, M. H., Hoque, M. R., Thapa, K. & Yang, S.-H., 2016. Two-Layer Hidden Markov Model for Human Activity Recognition in Home Environments. *International Journal of Distributed Sensor Networks*, Volume 2016.
- Keally, M. et al., 2011. *Pbn: towards practical activity recognition using smartphone-based body sensor networks.* s.l., s.n., pp. 246-259.
- Kearns, M., 1988. Thoughts on hypothesis boosting. *Unpublished manuscript*, Volume 45, p. 105.
- Kelleher, J., Namee, B. & D'Arcy, A., 2015. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies.* s.l.:MIT Press.
- Ke, S.-R. et al., 2013. A review on video-based human activity recognition. *Computers*, 2(2), pp. 88-131.

- Kim, Y., Schmid, T., Charbiwala, Z. M. & Srivastava, M. B., 2009. *ViridiScope: Design and Implementation of a Fine Grained Power Monitoring System for Homes*. New York, NY, USA, ACM, pp. 245-254.
- Kohavi, R. & John, G. H., 1997. Wrappers for feature subset selection. *Artificial intelligence*, 97(1), pp. 273-324.
- Kohavi, R. & others, 1995. *A study of cross-validation and bootstrap for accuracy estimation and model selection*. s.l., s.n., pp. 1137-1145.
- Kohavi, R. & Sommerfield, D., 1995. *Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology*. s.l., s.n., pp. 192-197.
- Krishnan, N. C. & Cook, D. J., 2014. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, Volume 10, Part B, pp. 138-154.
- Krishnan, N., Cook, D. J. & Wemlinger, Z., 2013. Learning a Taxonomy of Predefined and Discovered Activity Patterns. *J. Ambient Intell. Smart Environ.*, 5(6), pp. 621-637.
- Kumar, S. S. & John, M., 2016. *Human activity recognition using optical flow based feature set*. s.l., s.n., pp. 1-5.
- Kwapisz, J. R., Weiss, G. M. & Moore, S. A., 2011. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2), pp. 74-82.
- Larson, E. et al., 2012. Disaggregated water sensing from a single, pressure-based sensor: An extended analysis of HydroSense using staged experiments. *Pervasive and Mobile Computing*, Volume 8, pp. 82-102.
- Lawton, M. & Brody, E., 1988. *Instrumental Activities of Daily Living Scale (IADL)*. s.l.:s.n.
- Lee, J. W. et al., 2015. Persim 3D: Context-Driven Simulation and Modeling of Human Activities in Smart Spaces. *Automation Science and Engineering, IEEE Transactions on*, Oct, 12(4), pp. 1243-1256.

- Lee, J. W., Helal, A. (., Sung, Y. & Cho, K., 2014. *Context Activity Selection and Scheduling in Context-driven Simulation*. San Diego, CA, USA, Society for Computer Simulation International, pp. 9:1--9:8.
- Lee, J. W., Helal, A., Sung, Y. & Cho, K., 2013. *Context-Driven Control Algorithms for Scalable Simulation of Human Activities in Smart Homes*. s.l., s.n., pp. 285-292.
- Lee, S.-W. & Mase, K., 2002. Activity and location recognition using wearable sensors. *Pervasive Computing*, July-Sept, 1(3), pp. 24-32.
- Lester, J. et al., 2005. A hybrid discriminative/generative approach for modeling human activities.
- Lewin, D. et al., 2010. *Assisted living technologies for older and disabled people in 2030, A final report to Ofcom.*, s.l.: s.n.
- Li, J. et al., 2016. Feature Selection: A Data Perspective.
- Lin, W., Sun, M.-T., Poovandran, R. & Zhang, Z., 2008. *Human activity recognition for video surveillance*. s.l., s.n., pp. 2737-2740.
- Liu, H., Setiono, R. & others, 1996. *A probabilistic approach to feature selection-a filter solution*. s.l., s.n., pp. 319-327.
- Liu, J., Fujimaki, R. & Ye, J., 2013. Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint.
- Liu, S. et al., 2012. Multisensor Data Fusion for Physical Activity Assessment. *Biomedical Engineering, IEEE Transactions on*, March, 59(3), pp. 687-696.
- Logan, B. et al., 2007. *A Long-term Evaluation of Sensing Modalities for Activity Recognition*. Berlin, Heidelberg, Springer-Verlag, pp. 483-500.
- Luvstrek, M. & Kaluza, B., 2009. Fall detection and activity recognition with machine learning. *Informatica*, 33(2).

- Magnusson, L. & Hanson, E. J., 2003. Ethical issues arising from a research, technology and development project to support frail older people and their family carers at home. *Health & Social Care in the Community*, September, 11(5), pp. 431-439.
- Maimon, O. & Rokach, L., 2010. *Data Mining and Knowledge Discovery Handbook*. 2nd ed. s.l.:Springer Publishing Company, Incorporated.
- Maslow, A., 1987. Maslow's hierarchy of needs. *Salenger Incorporated*.
- Mendez-Vazquez, A., Helal, A. & Cook, D., 2009. *Simulating events to generate synthetic data for pervasive spaces*. s.l., s.n.
- Minnen, D. et al., 2006. Performance metrics and evaluation issues for continuous activity recognition. *Performance Metrics for Intelligent Systems*, Volume 4.
- Mitchell, T. M., 1997. *Machine Learning*. 1 ed. New York, NY, USA: McGraw-Hill, Inc..
- Murphy, K. P., 2012. *Machine learning: a probabilistic perspective*. s.l.:MIT press.
- Najafi, B. et al., 2003. Ambulatory system for human motion analysis using a kinematic sensor: Monitoring of daily physical activity in the elderly. *Ieee Transactions on Biomedical Engineering*, 50(6), pp. 711-723.
- Nef, T. et al., 2015. Evaluation of Three State-of-the-Art Classifiers for Recognition of Activities of Daily Living from Smart Home Ambient Data. *Sensors*, 15(5), pp. 11725-11740.
- Neumann, J., Schnorr, C. & Steidl, G., 2005. Combined SVM-based feature selection and classification. *Machine learning*, 61(1-3), pp. 129-150.
- Ng, A. Y., 2004. *Feature selection, L 1 vs. L 2 regularization, and rotational invariance*. s.l., s.n., p. 78.
- Nguyen, K., Stewart, R. & Zhang, H., 2013b. An intelligent pattern recognition model to automate the categorisation of residential water end-use events. *Environmental Modelling & Software*, Volume 47, pp. 108-127.

- Nguyen, K., Stewart, R. & Zhang, H., 2014. An autonomous and intelligent expert system for residential water end-use classification. *Expert Systems with Applications* , 41(2), pp. 342-356.
- Nguyen, K., Zhang, H. & Stewart, R., 2013a. Development of an intelligent model to categorise residential water end use events. *Journal of Hydro-environment Research*, 7(3), pp. 182-201.
- Nguyen, N., Bui, H., Venkatsh, S. & West, G., 2003. *Recognizing and monitoring high-level behaviors in complex spatial environments*. s.l., s.n., pp. II-620-5 vol.2.
- Noury, N. & Hadidi, T., 2012. Computer simulation of the activity of the elderly person living independently in a Health Smart Home. *Computer Methods and Programs in Biomedicine* , 108(3), pp. 1216-1228.
- Parkka, J. et al., 2006. Activity classification using realistic data from wearable sensors. *Information Technology in Biomedicine, IEEE Transactions on*, Jan, 10(1), pp. 119-128.
- Patel, S. N. et al., 2007. UbiComp 2007: Ubiquitous Computing: 9th International Conference, UbiComp 2007, Innsbruck, Austria, September 16-19, 2007. Proceedings. In: J. Krumm, G. D. Abowd, A. Seneviratne & T. Strang, eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 271-288.
- Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, Volume 12, pp. 2825-2830.
- Philipose, M. et al., 2004. Inferring activities from interactions with objects. *Pervasive Computing, IEEE*, Oct, 3(4), pp. 50-57.
- Pi, R., 2012. Raspberry pi. *Raspberry Pi*, Volume 1, p. 1.
- Plotz, T., Moynihan, P., Pham, C. & Olivier, P., 2011. Activity recognition and healthier food preparation. In: *Activity Recognition in Pervasive Intelligent Environments*. s.l.:Springer, pp. 313-329.
- Powers, D. M., 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

- Quinlan, J. R., 1986. Induction of decision trees. *Machine learning*, 1(1), pp. 81-106.
- Quinlan, J. R., 1993. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..
- Rashidi, P. & Cook, D. J., 2009. *Transferring Learned Activities in Smart Environments..* s.l., s.n., pp. 185-192.
- Rashidi, P. & Cook, D. J., 2010. *D.J.: Multi home transfer learning for resident activity discovery and recognition.* s.l., s.n., pp. 56-63.
- Riboni, D. & Bettini, C., 2009. *Context-aware activity recognition through a combination of ontological and statistical reasoning.* s.l., s.n., pp. 39-53.
- Richardson, I., 2015. KNX corner: The value of competition. *Connected Home Australia*, Issue Jun 2015, p. 77.
- Roy, P. C. et al., 2011. A possibilistic approach for activity recognition in smart homes for cognitive assistance to Alzheimer's patients. In: *Activity Recognition in Pervasive Intelligent Environments.* s.l.:Springer, pp. 33-58.
- Rozantsev, A., Lepetit, V. & Fua, P., 2015. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding* , Volume 137, pp. 24-37.
- Russell, D. J., 2010. Introduction to embedded systems: using ANSI C and the arduino development environment. *Synthesis Lectures on Digital Circuits and Systems*, 5(1), pp. 1-275.
- Salton, G. & McGill, M. J., 1983. Introduction to modern information retrieval.
- Sanchez, D., Tentori, M. & Favela, J., 2007. *Hidden Markov Models for Activity Recognition in Ambient Intelligence Environments.* s.l., s.n., pp. 33-40.
- Scanail, C. N. et al., 2006. A review of approaches to mobility telemonitoring of the elderly in their living environment. *Annals of Biomedical Engineering*, 34(4), pp. 547-563.
- Schapiro, R. E., 1990. The strength of weak learnability. *Machine learning*, 5(2), pp. 197-227.

- Schuller, B. & Burkhardt, F., 2010. *Learning with synthesized speech for automatic emotion recognition*. s.l., s.n., pp. 5150-5153.
- Sezer, E., Nefeslioglu, H. & Gokceoglu, C., 2014. An assessment on producing synthetic samples by fuzzy C-means for limited number of data in prediction models. *Applied Soft Computing* , Volume 24, pp. 126-134.
- Smirek, L., Zimmermann, G. & Ziegler, D., 2014. Towards universally usable smart homes-how can myui, urc and openhab contribute to an adaptive user interface platform. *IARIA, Nice, France*, pp. 29-38.
- Srinivasan, V., Stankovic, J. & Whitehouse, K., 2011. *WaterSense: Flow Disaggregation Using Motion Sensors*. s.l., s.n.
- Starner, T. & Pentland, A., 1995. *Real-time American Sign Language recognition from video using hidden Markov models*. s.l., s.n., pp. 265-270.
- Statnikov, A., Aliferis, C. F. & Hardin, D. P., 2011. *A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and Methods*. s.l.:world scientific.
- Steinhauer, H. J., Chua, S., Guesgan, H. & Marsland, S., 2010. *Utilising Temporal Information in Behavior Recognition*. s.l., s.n.
- Stikic, M., Huynh, T., Van Laerhoven, K. & Schiele, B., 2008. *ADL recognition based on the combination of RFID and accelerometer sensing*. s.l., s.n., pp. 258-263.
- Storf, H., Becker, M. & Riedl, M., 2009. s.l., s.n., pp. 1-7.
- Szewczyk, S. et al., 2009. Annotating smart environment sensor data for activity learning. *Technology and Health Care*, 17(3), p. 161.
- Tan, P.-N., Steinbach, M. & Kumar, V., 2005. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc..

- Tapia, E., Intille, S. & Larson, K., 2004. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. *Pervasive Computing, Lecture Notes in Computer Science*, Volume 3001, pp. 158-175.
- Tapia, E. M., Intille, S. S. & Larson, K., 2004. *Activity recognition in the home using simple and ubiquitous sensors*. s.l.:Springer.
- Turing, A. M., 1950. Computing machinery and intelligence. *Mind*, 59(236), pp. 433-460.
- Ubsdell, G., 2012. Wireless sound systems.
- Vail, D. L. & Veloso, M. M., 2008. *Feature Selection for Activity Recognition in Multi-Robot Domains*. s.l., s.n., pp. 1415-1420.
- Van Kasteren, T., Noulas, A., Englebienne, G. & Krose, B., 2008. *Accurate activity recognition in a home setting*. s.l., s.n., pp. 1-9.
- Varga, T. & Bunke, H., 2003. *Generation of synthetic training data for an HMM-based handwriting recognition system*. s.l., s.n., pp. 618-622.
- Vert, J.-P., Tsuda, K. & Scholkopf, B., 2004. A primer on kernel methods. *Kernel Methods in Computational Biology*, pp. 35-70.
- Wang, Y. Y. & Li, J., 2008. Feature-selection Ability of the Decision-tree Algorithm and the Impact of Feature-selection/Extraction on Decision-tree Results Based on Hyperspectral Data. *Int. J. Remote Sens.*, 29(10), pp. 2993-3010.
- Ward, J. A., Lukowicz, P. & Gellersen, H. W., 2011. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1), p. 6.
- Wen, J., Loke, S., Indulska, J. & Zhong, M., 2015. *Sensor-based activity recognition with dynamically added context*. s.l., s.n., pp. 1-10.
- Wilson, D. H. & Atkeson, C., 2005. Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. In: *Pervasive computing*. s.l.:Springer, pp. 62-79.

Witten, I. H., Frank, E. & Hall, M. A., 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..

Wonders, M., Ghassemlooy, Z. & Hossain, M. A., 2016. Training with synthesised data for disaggregated event classification at the water meter. *Expert Systems with Applications*, Volume 43, pp. 15-22.

Wong, W.-T. & Hsu, S.-H., 2006. Application of SVM and ANN for image retrieval. *European Journal of Operational Research*, 173(3), pp. 938-950.

Yamada, N. et al., 2007. Applying Ontology and Probabilistic Model to Human Activity Recognition from Surrounding Things. *Information and Media Technologies*, 2(4), pp. 1286-1297.

Young, P., 2015. *Political challenges relating to an aging population: Key Issues for the 2015 Parliament*. s.l.:House of commons library research.

Zhang, M. & Sawchuk, A., 2013. Human Daily Activity Recognition With Sparse Representation Using Wearable Sensors. *Biomedical and Health Informatics, IEEE Journal of*, May, 17(3), pp. 553-560.

Zhang, Z., 2012. Microsoft Kinect Sensor and Its Effect. *MultiMedia, IEEE*, Feb, 19(2), pp. 4-10.

Zhu, C. & Sheng, W., 2011. Wearable Sensor-Based Hand Gesture and Daily Activity Recognition for Robot-Assisted Living. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, May, 41(3), pp. 569-573.