

An End-to-End Framework for the Optimisation of Human Activity Recognition



Naomi Irvine

School of Computing
Faculty of Computing, Engineering
and the Built Environment
Ulster University

This dissertation is submitted for the degree of
Doctor of Philosophy

2021

(I confirm that the word count of this thesis is less than 100,000 words)

Table of Contents

Acknowledgements	VI
Abstract	VII
List of Figures	VIII
List of Tables	XI
Abbreviations	XIII
1 Introduction.....	1
1.1 Overview	1
1.2 Motivation for HAR Research	3
1.2.1 Pre-Processing	5
1.2.2 Feature Selection	5
1.2.3 Classification	6
1.3 Research Aim and Research Questions.....	6
1.4 Key Research Contributions	7
1.5 Thesis Structure.....	8
1.6 Publications	10
2 Literature Review	13
2.1 Overview	13
2.2 Human Activity Recognition	14
2.2.1 Sensor-based HAR	14
2.3 Application Domains	15
2.4 HAR Process	17
2.4.1 Data Acquisition	18
2.4.2 Pre-processing	22
2.4.2.1 Class-level Noise	22
2.4.2.2 Attribute-level Noise.....	23
2.4.3 Data Segmentation	24
2.4.4 Feature Extraction and Selection.....	26
2.4.4.1 Filter	28
2.4.4.2 Wrapper.....	29
2.4.4.3 Hybrid	31
2.4.5 Classification	32
2.5 Classification Algorithms	28

2.5.1 Generative Models	28
2.5.2 Discriminative Models	30
2.5.3 Ensemble Methods	34
2.5.3.1 Ensemble Generation	36
2.5.3.2 Ensemble Integration	38
2.6 HAR Challenges	40
2.6.1 Intraclass Variability	40
2.6.2 Interclass Similarity.....	41
2.6.3 Class Imbalance.....	42
2.6.4 Recognition of Interleaved and / or Concurrent Activities	44
2.7 Conclusion	46
3 The Impact of Dataset Quality on the Performance of Data-Driven Approaches to HAR.....	48
3.1 Overview	48
3.2 Data Quality	49
3.3 Data Cleansing	50
3.3.1 Outliers	50
3.3.2 Missing Values	52
3.3.3 Inconsistent or Incorrect values.....	53
3.4 Methodology	54
3.4.1 Data Acquisition.....	55
3.4.2 Data Cleaning.....	58
3.4.3 Segmentation.....	62
3.4.4 Feature Extraction	62
3.4.5 Activity Classification.....	64
3.5 Results and Discussion.....	64
3.6 Conclusion	68
4 Recommendations for Pre-Processing Publicly Available HAR Datasets	70
4.1 Overview	70
4.2 Dataset for Data-Driven HAR.....	71
4.2.1 UCAmI Cup Dataset	72
4.2.2 Data Challenges.....	75
4.2.3 Acknowledged Limitations	76
4.2.4 Restructured Dataset of Improved Quality.....	79

4.3 Data Pre-Processing	80
4.4 Recommendations	81
4.5 Conclusion	84
5 Selecting an Optimal Subset of Features	86
5.1 Overview	86
5.2 Methodology	87
5.2.1 Information Gain	88
5.2.2 Correlation.....	89
5.2.3 Relief-F.....	90
5.2.4 Sequential Selection: Forward and Backward	92
5.2.5 Rationale for Feature Selection Threshold.....	94
5.3 Initial Experimental Results	97
5.4 Hybrid-Filter Approach.....	100
5.4.1 Rationale.....	100
5.4.2 Methodology	101
5.4.3 Results and Discussion.....	103
5.5 Conclusion	106
6 Homogeneous Neural Network Ensemble for Human Activity Recognition	108
6.1 Overview	108
6.2 Rationale for Homogeneous NN Ensemble	109
6.3 Methodology	110
6.3.1 Proposed HAR Classification Model.....	110
6.3.1.1 Complement Class Generation at a Model Level	116
6.3.1.2 Complement Class Generation at a Class Level.....	117
6.3.2 Model Conflict Resolution	118
6.4 Results and Discussion.....	121
6.5 Conclusion	125
7 Heterogeneous Ensembles for Human Activity Recognition	127
7.1 Overview	127
7.2 Rationale for Heterogeneous Ensemble	128
7.3 Proposed Heterogeneous Ensemble Methods	130
7.3.1 Ensemble Method 1.....	132
7.3.2 Ensemble Method 2.....	135

7.4 Results and Discussion.....	143
7.5 Conclusion	150
8 Conclusions and Future Work.....	152
8.1 Overview	152
8.2 Discussion	153
8.3 Summary of Contributions	155
8.4 Limitations	160
8.5 Proposed Future Work	161
8.6 Conclusion and Future Direction	163
References	164
Appendix 1	182
Appendix 2	191

Acknowledgements

I would like to take this opportunity to express my sincere appreciation to those who have provided their continuous encouragement and support throughout my PhD journey. Primarily, I would like to thank my supervisory team within Ulster University Professor Chris Nugent, Dr Shuai Zhang and Professor Hui Wang, and my external supervisor Professor Wing Ng of Southern China University of Technology. Their support, guidance and encouragement throughout my PhD studies has been incredible, and for that I am truly grateful. I would also like to thank them for the overwhelming opportunities they have given me throughout my PhD, and for providing their advice and feedback during each of my studies.

I would like to express my gratitude to the PhD researchers within Ulster University that have become close friends. Their advice, humour and encouragement has been amazing, and I'm so thankful for all the memories. Particularly, the hilarious conversations over lunch and tea breaks.

For their enduring support throughout life, I would like to thank my family. My mother Lisa, my father Alex, and my brothers James, Thomas and Michael. I owe so much to you all and thank you for always believing in me. Finally, I would like to thank Mark Walker for his love and support, and for always making me smile even through the difficult days.

Abstract

Due to recent advancements and the incessant progression of wireless sensor networks, conducting Human Activity Recognition (HAR) research within smart environments has become a widely explored domain. Nevertheless, whilst extensive research has been carried out, HAR remains a highly intricate and challenging task. Each stage of the data-driven HAR process contributes to the overall performance, thus, optimisation within each stage has driven research endeavors.

This Thesis presents an end-to-end methodology for the optimisation of HAR, which involves investigations into enhancing performance at various key stages of the process. A publicly available HAR dataset was utilised throughout to evaluate and demonstrate the effectiveness of the proposed approach.

Initial explorations focused upon the pre-processing stage, within which the impact of data quality upon activity classification was explored using data-driven approaches to HAR. Findings demonstrated the negative impact of noise upon classification performance, with a significant performance increase of 12.97% when using cleaned data. This work led to providing recommendations as to how data should be pre-processed to prevent reductions in performance. Subsequent explorations focused upon enhancing HAR performance during the feature selection stage, within which a new hybrid feature selection method was produced. Findings revealed the effectiveness of the developed method which achieved an enhanced HAR performance of 83.24%, in addition to demonstrating the benefits of performing feature selection. A considerable trade-off was revealed between the classification performances achieved and the number of redundant features identified and removed, in comparison to the evaluated well-established feature selection techniques. Finally, research endeavours focused upon optimising HAR performance during the classification stage, within which both novel homogeneous and heterogeneous ensemble methods were produced. Findings demonstrated the effectiveness of the proposed ensembles, in particular the heterogeneous method which outperformed 4 benchmarked classifiers achieving an overall classification performance of 84.13%.

List of Figures

Figure 1.1. The HAR process, adapted from [16], which highlights key areas explored within this Thesis.	4
Figure 1.2. Outline of research studies undertaken and their resulting publications	12
Figure 2.1. Smart environment which involves sensors, data processing and actuators [65]	16
Figure 2.2. The HAR process which includes data acquisition, pre-processing, data segmentation, feature extraction and selection, and classification, adapted from [17]	18
Figure 2.3. Accelerometer signals produced through performing static and dynamic activities	19
Figure 2.4. Environmentally deployed sensors, including contact switches, motion detectors and pressure sensors [79]	20
Figure 2.5. Examples of time and event-based windowing on binary data, where a 15 second time window is displayed, along with an event-based window comprising 15 events, adapted from [85]	25
Figure 2.6. The Filter Process [20]	28
Figure 2.7. The Wrapper Process [20]	29
Figure 2.8. Example of class imbalance existing within HAR data produced for the UCAmI Cup [79]	42
Figure 2.9. Examples of interleaved and concurrently performed activities, adapted from [157]	45
Figure 3.1. Shimmer device where (a) presents the Shimmer device axis, and (b) presents an example of the correct Shimmer device placement on a wrist.	55
Figure 3.2. Shimmer Calibration software	56
Figure 3.3. (a) presents an example of the Shimmer Connect software used to record activity data from the accelerometer and (b) presents a full example of the produced activity recording (approximately 2 minutes duration)	57
Figure 3.4. Data cleaning process within which sources of error (noise) are identified and handled	59

Figure 3.5. Displaying a signal before and after the removal of noise, caused by a time delay at the beginning of a recording	60
Figure 3.6. Random large spike to be removed from the signal	61
Figure 3.7. Displaying a signal with a range outside the measurable capability	61
Figure 4.1. Location of binary sensors in the UJAmI Smart Lab [79].....	73
Figure 4.2. Distribution of the 24 UCAmI Cup activity classes with threshold of <30 instances presented as dotted line.	80
Figure 4.3. Excerpt from a raw binary data file	81
Figure 5.1. Output values produced by Information Gain	89
Figure 5.2. Output values produced by Correlation	90
Figure 5.3. Relief-F concept illustrating the neighbour selection [94]	91
Figure 5.4. Output values produced by Relief-F	92
Figure 5.5. Output produced by Wrapper method	93
Figure 5.6. Filter feature pools demonstrating the AND and XOR features.....	102
Figure 6.1. Four base classifiers presented per time routine, where n indicates the number of classes per model. M, A, and E represent the Morning, Afternoon, and Evening models, respectively, and finally MI represents the Mixed model.....	111
Figure 6.2. Framework for the homogeneous ensemble approach. M_1 , M_2 and M_3 represent the Morning, Afternoon and Evening models, respectively, and M_4 represents the Mixed model.	115
Figure 6.3. Human Activity Recognition (HAR) performance per conflict resolution approach.	122
Figure 6.4. HAR performance of the proposed ensemble NN approach compared to kNN, SVM, NN, and Logistic Regression classifiers, in terms of accuracy (%). .	125
Figure 7.1. The heterogeneous ensemble generation process for M_1 , where n indicates the number of classes per model.	131
Figure 7.2. Heterogeneous Ensemble Method 1	132
Figure 7.3. Heterogeneous Ensemble Method 2	136
Figure 7.4. Majority Voting implementation where the class outputs from each base classifier are passed into the majority voting module to ascertain the final voted output class.....	137
Figure 7.5. Weighted Majority Voting implementation where W represents the weights being applied to the outputs of each base classifier, and n represents the number of classes per model.	138

Figure 7.6. Phase 2 of the proposed heterogeneous approach, within which new conflict resolution algorithms were implemented.....	142
Figure 7.7. HAR performances achieved through Heterogeneous Ensemble Method 1.....	144
Figure 7.8. HAR performances achieved through Heterogeneous Ensemble Method 2.....	146
Figure 7.9. Comparisons of each ensemble method, including the homogeneous method, and heterogeneous methods 1 & 2	149
Figure 8.1. The HAR process, adapted from [16], which highlights, in red, the key areas explored within this Thesis	156

List of Tables

Table 2.1. Identified HAR datasets collected through body-worn and environmental sensors	21
Table 2.2. Data-driven classification algorithms	34
Table 3.1. Dataset description including the number of participants assigned to each HAR scenario, and the activities involved within each of the 6 scenarios	54
Table 3.2. Features extracted from the windowed data.....	63
Table 3.3. Accuracies of four algorithms for the classification of activities included in the Self-Care scenario.	65
Table 4.1. Activity classes in the UCAmI Cup dataset [60], where M, A and E indicate the Morning, Afternoon and Evening routines, respectively.	73
Table 4.2. Description of binary sensors [35]	74
Table 4.3. UCAmI Cup Challenge: Implemented techniques, performances achieved, and data challenges reported by participants	76
Table 4.4. Activities producing similar sensor characteristics within the UCAmI Cup data	78
Table 4.5. Activity Classes in the Restructured Dataset, where underrepresented activities in the original dataset have been either removed or merged	80
Table 4.6. Recommendations for pre-processing HAR data.....	82
Table 5.1. Features considered in each experiment, where Y indicates inclusion in the subset and N indicates removal of the feature.....	96
Table 5.2. No feature selection applied. All original 31 features are included.	97
Table 5.3. Feature selection applied via Wrapper techniques, where the values in brackets indicate the comparative difference between the current performance and the performances achieved with no feature selection applied.....	98
Table 5.4. Feature selection applied via Filter techniques, where the values in brackets indicate the comparative difference between the current performance and the performances achieved with no feature selection applied.....	99
Table 5.5. Newly generated feature subsets based upon combined filters.....	102
Table 5.6. Classification performances based on newly generated feature subsets, where the values in brackets indicate the comparative difference between the current performance and the performances achieved with no feature selection applied....	104

Table 5.7. Comparison of all features, Relief-F and Subset 6.....	105
Table 6.1. Activity class outputs per model.	112
Table 6.2. Model level-distribution of instances for complement class compositions, where M, A, E, and MI within the class distributions indicate classes belonging to the Morning, Afternoon, Evening, and Mixed models, respectively.	117
Table 6.3. Class level-distribution of instances for complement class compositions, where M, A, E, and MI within the class distributions indicate classes belonging to the Morning, Afternoon, Evening, and Mixed models, respectively.	118
Table 6.4. Number of conflicts occurring, per fold, through each data distribution of the complement class.	122
Table 6.5. Ensemble approach 1 - Analysis of incorrect instances, where A.6.2, A.6.3, A.6.4 and A.6.5 represent the Algorithm number	123
Table 6.6. Ensemble approach 2 – Analysis of incorrect instances, where A.6.2, A.6.3, A.6.4 and A.6.5 represent the Algorithm number	124
Table 7.1. Training performances achieved by each base classifier	133
Table 7.2. Example of a conflict occurring between Base Models M_2 and M_3	139
Table 7.3. Conflicts occurring between base models	143
Table 7.4. Ensemble method 1 - Analysis of incorrect instances.....	146
Table 7.5. Ensemble method 2 via majority voting – Analysis of incorrect instances	148
Table 7.6. Ensemble method 2 via weighted majority voting – Analysis of incorrect instances	148

Abbreviations

AAL	Ambient Assisted Living
ADL	Activity of Daily Living
CFS	Correlation Feature Selection
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Network
DT	Decision Tree
ECOC	Error-Correcting Output Code
HAR	Human Activity Recognition
HMM	Hidden Markov Model
kNN	k-Nearest Neighbour
LR	Logistic Regression
MLP	Multilayer Perceptron
NB	Naïve Bayes
NN	Neural Network
OVA	One-vs-All
OVO	One-vs-One
PIR	Passive Infrared
RFID	Radio-Frequency Identification
RMS	Root Mean Square
RNN	Recurrent Neural Network
SBS	Sequential Backward Selection
SFS	Sequential Forward Selection
SMA	Signal Magnitude Area
SMV	Signal Magnitude Vector
SVM	Support Vector Machine
WHO	World Health Organisation

Chapter 1

Introduction

1.1 Overview

Demographic concerns pertaining to the ageing population have been recognised by the World Health Organisation (WHO) stating that “In 2019, the number of people aged 60 years and older was 1 billion, and this number will increase to 2.1 billion by 2050” [1]. The prevalence of health decline amongst the ageing generation, such as the emergence of chronic illnesses and disability, has contributed to the continually increasing cost of healthcare provision, whilst also placing greater strain upon healthcare providers due to staff shortages [2]. Thus, due to the continually increasing healthcare costs, the need for an alternative cost-effective means of providing care has emerged.

The demographic issues outlined, in addition to the prevalence of health decline amongst the ageing population, have formed a need to promote the concept of “ageing in place” to enhance quality of life of the ageing generation [2]. “Ageing in place” supports the ageing in that they may remain in their own homes for longer, living independently, which is vital in protecting their mental health as well as decelerating their physical decline as opposed to residing in an institutionalised care facility [3]. According to [4], a large portion of the ageing population would prefer to remain in their own homes rather than living in dedicated care facilities, thus supportive measures need to be further developed to support such desires.

The specified concerns are being addressed through a large number of solutions, one of which involves conducting smart environment research in relation to ambient intelligence and the production of assistive technologies. Ambient Assisted Living (AAL) is a technology focused approach aimed at supporting independent living and enhancing the wellbeing of inhabitants. These technologies may be beneficial in promoting and improving the self-management of health issues, for example, through providing reminders for inhabitants to take their medication and through monitoring and supporting their mobility within their home setting [4]. Furthermore, AAL technologies may promote the safety of smart environment inhabitants, for example, through fall detection implementations, as fall risk of the ageing is particularly prevalent, often leading to significant injuries [4]. Nevertheless, a number of acknowledged concerns regarding AAL technology adoption have emerged. For example, even though many of the ageing population are willing to attempt the use of such new technologies, they often lack understanding and confidence to effectively use them, thus obstructing continued adoption [3]. Privacy concerns regarding AAL technologies also restrict their adoption, for example, many of the ageing population are hesitant of the installation of visual-based solutions [5]. Alternatively, sensor-based solutions may be deployed to address privacy issues. For example, environmental sensors may be deployed within the home to monitor smart environment inhabitants unobtrusively [5].

1.2 Motivation for HAR Research

Human Activity Recognition (HAR) can be defined as the ability to recognise and interpret human activity automatically through the deployment of sensors and subsequent processing of the collected sensor data [6]. It is a dynamic and challenging research area [7] as human activities are intricate, highly diverse and also person specific, for example walking styles can largely vary in terms of speed and gait length depending on various factors, for example age [8]. The monitoring of Activities of Daily Living (ADL) within smart environments is a significant consideration for assessing the health status of inhabitants, thus the automatic detection of these activities is the core motivation for conducting HAR research [9]. ADLs are considered as an assessment of wellbeing, where an inhabitant's cognitive and/or physical abilities to independently accomplish basic activities are evaluated, for example, preparing a meal, personal grooming, dressing, and taking medication. The ability to perform ADLs are essential in ensuring a person can reside and function adequately within their home setting [10], [11]. Thus, HAR research is essential in improving AAL technologies to assist with independent living, and consequently improving the quality of life of smart environment inhabitants, in addition to alleviating some of the burden placed upon care providers.

Due to advancements and the continuous progression of unobtrusive wireless sensor networks, activity monitoring within smart environments has become common within which sensors are deployed to collect information [12]. Nevertheless, a critical concern has been identified in that no cohesive standards exist for the collection and formatting of sensor data [12], which has led to a scarcity in high quality, publicly available datasets. This lack of available data continues to hinder HAR research, particularly within the realms of data-driven approaches as these rely on good quality data for classification [13]. Thus, the need to develop clear data collection and storage standards exists to encourage researchers to effectively generate and disseminate high quality datasets to support the research community in evaluating their approaches to HAR.

The HAR process normally consists of 5 fundamental stages for data-driven approaches, including pre-processing, data segmentation, feature extraction

and selection and classification. Whilst extensive research has been conducted within these areas in recent years [14], HAR remains a challenging task with vast scope for further investigations and improvement [12]. This Thesis considers various vital stages of the HAR process, as presented in Figure 1.1, which highlights the pre-processing stage including data cleaning, scrubbing and wrangling, feature extraction and selection, and classification. Data cleaning and scrubbing refers to the process of removing inaccurate data, whereas wrangling typically involves transforming the format of data, for example, through achieving consistent naming conventions [15]. Particularly, discovering methods of enhancing performance at each of the identified stages is the key motivation driving their exploration, as according to [16] each stage of the HAR process contributes an effect upon the overall performance. Within the following Sections, rationale has been identified as to how performance enhancements can be achieved at each stage.

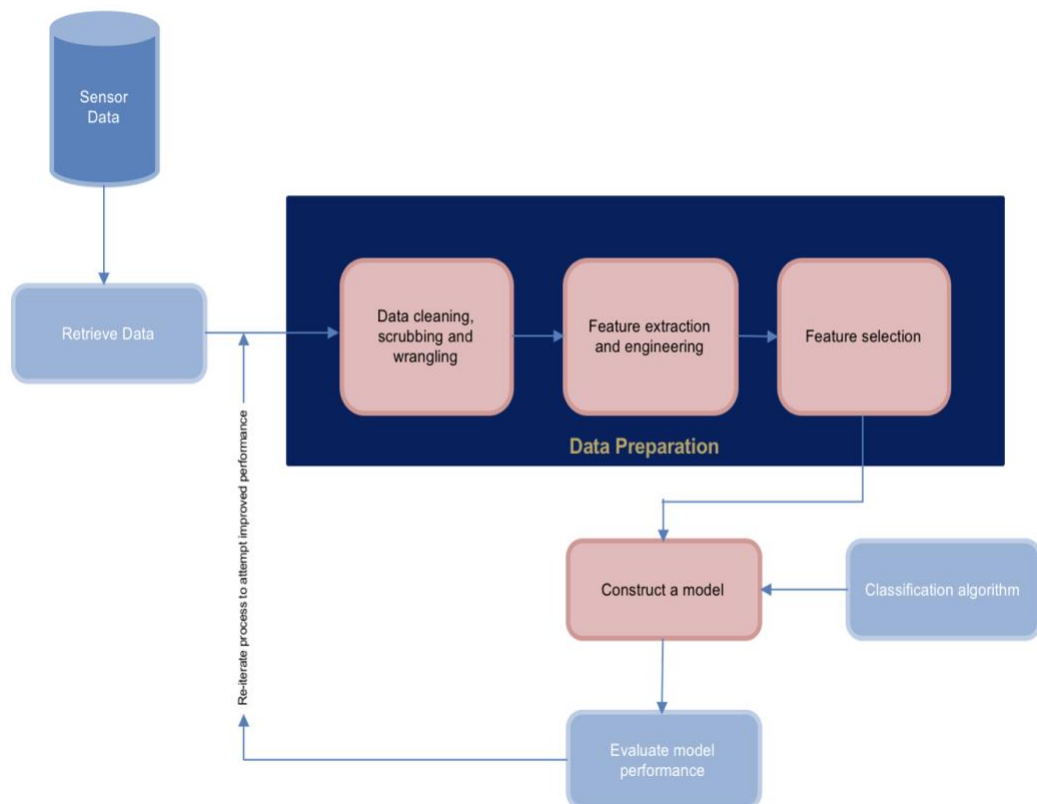


Figure 1.1 - The HAR process, adapted from [16], which highlights key areas explored within this Thesis.

1.2.1 Pre-Processing

During the data pre-processing stage which involves data cleaning, scrubbing and wrangling, data is prepared for further processing by reducing noise and data redundancy [7]. The reduction/removal of noise is an important consideration within HAR, as noise can negatively impact upon classification performance [17]. Two types of noise may transpire, namely attribute noise and class noise [18]. Attribute noise may be introduced during data collection where one or more attribute values become erroneous or corrupt, whereas class noise may be introduced during labelling and annotation, for example, mislabelling data instances [18]. According to [19], attribute noise is more harmful and more challenging to address, however, research into attribute noise has been notably neglected in recent years, thus achieving little progression [19]. This has provided the impetus for investigations into the effects of attribute noise, as good quality data is essential for optimal classification performance [18].

1.2.2 Feature Selection

Within the feature selection stage of the HAR process, the key objective is to discover an optimal subset of features capable of effectively distinguishing between various activities during classification. Particularly, within the realms of environmentally deployed sensors within smart environments, feature selection provides additional value in that redundant sensors can be identified and removed, thus reducing costs. It has been recognised that the discovery and removal of irrelevant features can improve classification performance [20], thus emphasising the importance of attentively considering the feature extraction and selection stage of the HAR process. Conventional approaches, such as applying filters and wrappers, have proven beneficial within many recent studies [21]–[25]. Nevertheless, hybrid methods have more recently been attracting research interest which involve combining conventional approaches, for example hybrid filter-wrapper techniques, and have subsequently demonstrated promising results [26]–[29], thus indicating scope for further investigations.

1.2.3 Classification

Classification is the final stage of the HAR process, within which the previously selected features are used as inputs to a classification model. Numerous well-established classification algorithms exist and have demonstrated their effectiveness when applied to HAR problems, for example Support Vector Machine (SVM) [30]–[33], Decision Tree (DT) [34]–[37], Naïve Bayes (NB) [38]–[40], k-Nearest Neighbour (kNN) [41]–[43] and Neural Networks (NN) [44]–[46]. Particularly, NNs have attracted interest for HAR tasks due to their predictive performance and their ability to model complex, non-linear relationships which is a valuable quality within the HAR domain [47].

Nevertheless, ensemble methods have been explored more recently which involve the combination of conventional classifiers due to their perceived effectiveness in further improving classification performance, which have recently demonstrated success [48], [49]. According to [50], the limitations of individual classifiers can be diminished through effectively combining multiple classifiers within an ensemble method, thus further enhancing classification performance and generalisation capabilities. Diversity has been recognised as an essential consideration in generating an effective ensemble method through achieving dissimilar decisions, which has been explored at both data and/or classifier levels [51]. Nevertheless, according to [51] diversity within ensemble methods has scope for further investigations. For example, through diversifying the inputs to base classifiers and/or diversifying the chosen classifiers through employing different classification algorithms.

1.3 Research Aim and Research Questions

Section 1.2 identified various areas within the HAR process providing opportunity for further research investigations. Through improving each of these areas, the overall HAR performance will be collectively improved, thus forming motivation for research endeavours within this Thesis.

Aim

The aim of this research is to improve the performance of HAR within smart environments.

Research Questions:

- 1 What are the research challenges associated with HAR that may hinder classification performance?
- 2 To what extent does data quality impact upon HAR performance?
- 3 Can hybrid feature selection methods offer additional benefits in producing an optimal subset of features for HAR?
- 4 Can generating diversity within ensemble methods effectively enhance HAR performance?

1.4 Key Research Contributions

Following the motivation behind the set of Research questions identified, an end-to-end framework for the optimisation of HAR is presented within this Thesis. This framework focusses on optimising performance at various fundamental stages of the HAR process to accomplish the aim of this Thesis of improving the overall HAR performance. The areas focussed upon within this Thesis include pre-processing, feature extraction and selection and classification, as previously presented in Figure 1.1. A publicly available dataset, containing ADLs performed within a smart environment, is explored at the previously identified stages of the HAR process within which various methods are investigated in an endeavour to optimise performance. This Thesis provided the following contributions to knowledge:

- Recommendations for pre-processing data to improve the performance of data-driven approaches to HAR.
- Produced a new approach to select an optimal subset of features for HAR.
- A new homogeneous ensemble classification model that introduces diversity at a data level.

- A new heterogeneous ensemble classification model that introduces diversity at both data and classifier levels.

1.5 Thesis Structure

The remaining 7 Chapters of this Thesis are summarised as follows:

- Chapter 2: Literature Review

This Chapter presents a comprehensive review of the literature pertaining to HAR research in an attempt to identify future trends and research opportunities. This Chapter involves a review of the typical 5 stage HAR process, including data acquisition, pre-processing, data segmentation, feature extraction and selection and classification. Various application domains for HAR are identified and various challenges associated with HAR research are discussed. Furthermore, a range of classification algorithms are evaluated.

- Chapter 3: The Impact of Dataset Quality on the Performance of Data-Driven Approaches to HAR

This Chapter considers the impact of data quality upon activity classification using data-driven classification models. Various classification algorithms were applied to produce models for HAR, where the importance of data quality was emphasised by evaluating the effects of noisy data through generating comparisons between the performances of raw and subsequently cleaned, HAR data. Experimental results obtained within this Chapter demonstrated the adverse impact of noise upon classification performance, as well as emphasising the importance of attentively adhering to a data collection protocol to diminish the introduction of noise.

- Chapter 4: Recommendations for Pre-Processing Publicly Available HAR Datasets

This Chapter involves assessing, and improving, the quality of an openly available HAR dataset for the purpose of data-driven HAR through identifying a range of data challenges and subsequently restructuring the data to enhance its quality for further processing. Recommendations for pre-processing data are also provided, based upon the findings from Chapters 3 and 4. The findings within this Chapter support and reinforce the need for investigations into the development of unified standards pertaining to data collection and sharing, which will aid the progression of HAR research. The data introduced within this Chapter is processed throughout all subsequent Chapters, thus providing an end-to-end methodology.

- Chapter 5: Selecting an Optimal Subset of Features

This Chapter investigates the impact of various well-established filter and wrapper feature selection methods upon binary sensor-based HAR data. This Chapter also involves explorations into a new hybrid feature selection method to ascertain an optimal subset of features for the purpose of HAR in an endeavor to enhance classification performance. Findings within this Chapter demonstrate the effectiveness of the explored hybrid feature selection method, where a considerable trade-off has been recognised between the classification performance obtained and the number of redundant features removed when compared to the initially explored filter and wrapper methods. This Chapter ultimately demonstrates the benefits of performing feature selection, as all explored methods demonstrate enhanced HAR performance in comparison to no feature selection being applied.

- Chapter 6: Homogeneous Neural Network Ensemble for Human Activity Recognition

Within this Chapter, a new homogeneous ensemble of NNs is explored, as well as various methods to resolving conflicts that may occur between base models in ensemble classifiers that have been trained on unique classes. This Chapter explores diversity at a data level only through diversifying the inputs to each NN base classifier. Two data distributions are explored in generating the complement

class per base model, which involve distributing data at a class level or a model level, and various conflict resolution techniques are explored. Findings within this Chapter demonstrate the effectiveness of the class level data distribution technique to effectively generate a complement class per base model. It is also found that the proposed homogeneous NN ensemble outperforms two of the four benchmarked classifiers.

- Chapter 7: Heterogeneous Ensembles for Human Activity Recognition

This Chapter further explores ensemble classifiers. Previously, homogeneous ensemble classifiers were explored, thus, this Chapter involves explorations of heterogeneous ensemble classifiers to investigate diversity further. Within this Chapter, diversity has been achieved at both a data and classifier level through additionally selecting and generating diverse base classifiers, whereas within the previous Chapter, diversity was achieved at a data level only. Two diverse heterogeneous ensemble methods are explored within this Chapter, with experimental findings demonstrating the effectiveness of the second method. Both heterogeneous ensembles outperform the homogeneous NN ensemble, indicating that diversity introduced amongst diversifying the base classifiers is an effective approach. Generating heterogeneous ensembles also outperformed all four benchmarked classifiers, thus achieving the most superior HAR performance.

- Chapter 8: Conclusion and Future Work

This Chapter discusses the key findings and conclusions generated throughout this Thesis. A summary of the key research contributions is presented and discussed, as well as outlining limitations encountered and further research areas to explore.

1.6 Publications

The following are a list of publications associated with this PhD:

- N. Irvine, C. Nugent, S. Zhang, H. Wang, and W. Y. NG. “Neural Network Ensembles for Sensor-Based Human Activity Recognition Within Smart Environments”, *Wearable and Unobtrusive Biomedical Monitoring, Sensors* (2020), 20(1), 216. <https://doi.org/10.3390/s20010216>. [Journal contribution]
- N. Irvine, C. Nugent, S. Zhang, H. Wang, W. Y. NG, I. Cleland, and M. Espinilla. “The Impact of Dataset Quality on the Performance of Data-driven Approaches for Human Activity Recognition”, *Proceedings of the 13th International FLINS Conference on Data Science and Knowledge Engineering for Sensing Decision Support*, pp. 1300-1308, (2018). [International conference]
- M. Espinilla, J. Medina, N. Irvine, I. Cleland, and C. Nugent. “Fuzzy Framework for Activity Recognition in a Multi-Occupant Smart Environment based on Wearable Devices and Proximity Beacons”, *Proceedings of the 13th International FLINS Conference on Data Science and Knowledge Engineering for Sensing Decision Support*, pp. 1292-1299, (2018). [International conference]
- M. Espinilla, J. Medina, A. Salguero, N. Irvine, M. Donnelly, I. Cleland, C. Nugent. “Human Activity Recognition from the Acceleration Data of a Wearable Device. Which Features Are More Relevant by Activities?” *Proceedings of the 12th International Conference on Ubiquitous Computing and Ambient Intelligence*, 2(19), 1242, (2018). [International conference]
- S. Zhang, W. Y. NG, J. Zhang, C. Nugent, N. Irvine, T. Wang. “Evaluation of Radial Basis Function Neural Network Minimising L-GEM for Sensor-based Activity Recognition”, *Journal of Ambient Intelligence and Humanized Computing*, (2019). <https://doi.org/10.1007/s12652-019-01246-w>. [Journal contribution]

The “Neural Network Ensembles for Sensor-Based Human Activity Recognition Within Smart Environments” publication was based upon the methodology and results obtained through developing a novel homogeneous NN ensemble classifier within Chapter 6 of this Thesis. Furthermore, “The Impact of Dataset Quality on the Performance of Data-driven Approaches for Human Activity Recognition” publication was based upon the findings within Chapter 3.

The remaining publications listed underpinned the work involved with this Thesis. They were representative of contributions made within a collaborative endeavor along with external researchers. However, through gaining the opportunity to collaborate with external researchers throughout the duration of this PhD, relevant and valuable domain knowledge and experience was acquired which supported the design of methodologies within Chapters 3, 5, 6 and 7. Figure 1.2 presents an outline of the research studies undertaken and their produced publications.

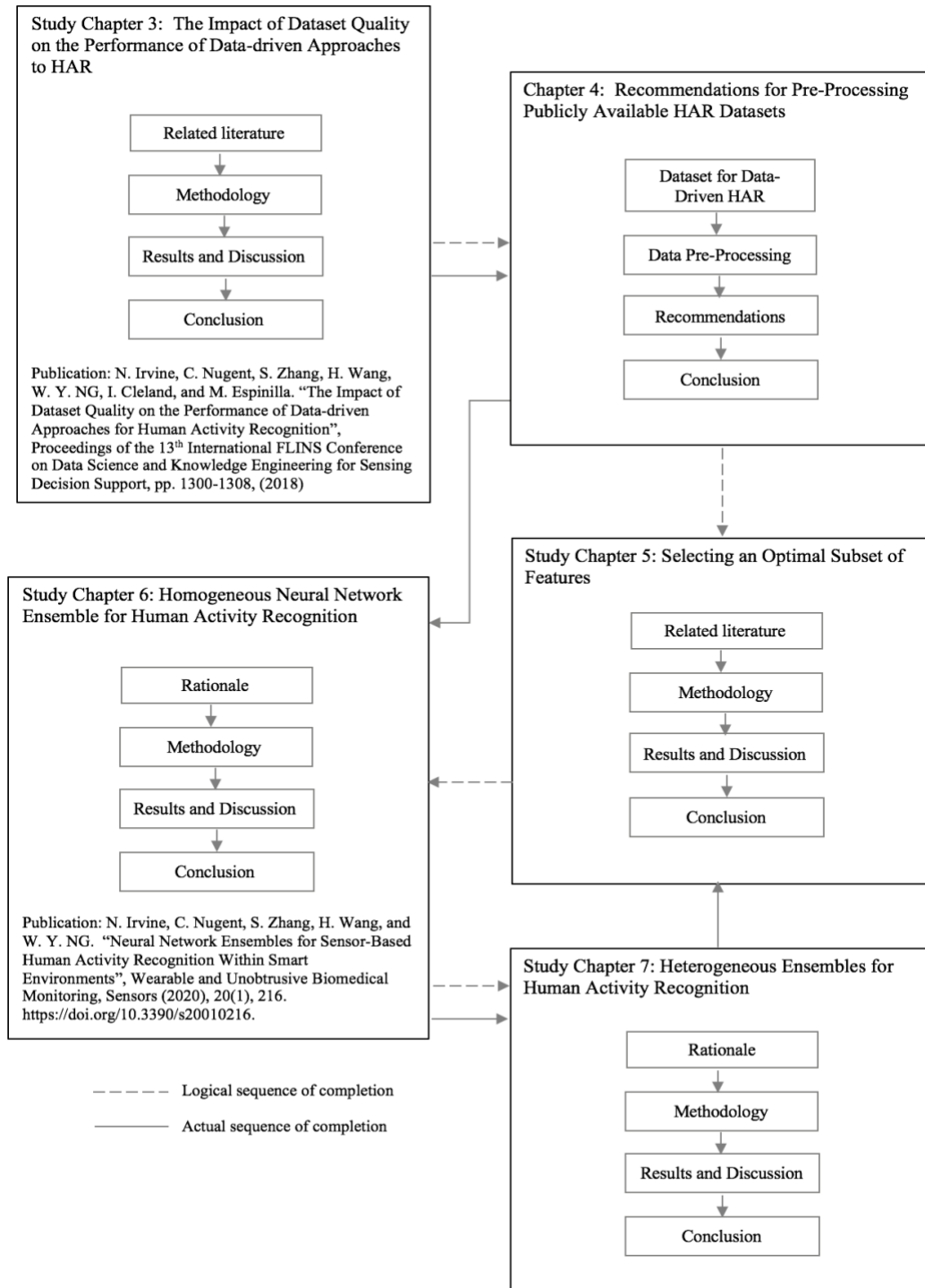


Figure 1.2. Outline of research studies undertaken and their resulting publications

Chapter 2

Literature Review

2.1 Overview

A comprehensive review of the literature involving HAR is presented in this Chapter. Discussions include a review of the HAR process from the perspective of 5 fundamental stages, namely data acquisition, pre-processing, data segmentation, feature extraction and selection, and classification. The various application domains for HAR are also discussed, in addition to the identified challenges associated with conducting HAR investigations.

Section 2.2 involves an overview of HAR. Section 2.3 introduces HAR application domains, Section 2.4 describes each stage of the HAR process, Section 2.5 describes various classification algorithms, and Section 2.6 describes challenges which have been openly identified. Finally, Section 2.7 concludes this Chapter.

2.2 Human Activity Recognition

HAR can be generally deemed within two categories of deployed technologies: either sensor-based or vision-based activity recognition [52]. Particularly, due to advancements with wireless sensor networks and sensing technologies, sensor-based activity recognition has attracted considerable research interest [53]. It is promising within health and wellbeing application domains as sensor-based approaches can better address privacy and ethical issues, in comparison to vision-based approaches to HAR that can be deemed obtrusive to users [54]. Furthermore, it has been recognised that vision-based approaches are limited in that blurred motion may occur, visual angle and path obstruction issues may hinder data gleaning, and fluctuating illumination occurs within complex environments [52].

2.2.1 Sensor-based HAR

There are generally two categories of sensor-based activity recognition: data-driven and knowledge-driven approaches [54]. Data-driven approaches learn activity models through the use of data mining and machine learning techniques with large-scale datasets [54]. Their ability to handle uncertainty and temporal information is deemed advantageous, though large-scale datasets are required for their implementation and the learnt models are generally difficult to apply to a range of people, resulting in reusability concerns [54]. Nevertheless, the large-scale datasets required to implement data-driven approaches are often difficult to obtain, as a shortage in publicly available, high quality, annotated datasets has been recognised as a challenge that continues to hinder HAR research [13]. Alternatively, knowledge-driven approaches exploit rich prior knowledge in the domain of interest to build activity models [55]. Reusability issues are eliminated through the implementation of knowledge-driven techniques as the models built are generic, thus they can be applied to various users. Nevertheless, models from knowledge-driven approaches are weak at handling uncertainty and temporal information [55], in addition to often possessing a weak ability in handling complex activity data as providing predefined human knowledge upon simplistic activities, which involve the basic steps required to perform activities, can be insufficient in representing and

capturing their fine-grained components [56]. Furthermore, there is a low availability of domain knowledge provided by experts. Another approach has been emerging in recent years, namely the hybrid method, which incorporates both data-driven and knowledge-driven techniques. The aim of this combined method is to overcome identified limitations of individual approaches, for example, an acknowledged limitation of knowledge-driven techniques is that the models built are static in that they cannot adjust to user preferences automatically, and it is challenging to define comprehensive activity models [57]. Thus, through combining data-driven with knowledge-driven approaches, the activity models produced are capable of continuously adjusting and learning various user preferences [57]. For example, a hybrid technique proposed in [55] demonstrates the combined method by primarily generating knowledge-based activity models which contain only the basic, essential steps required to complete certain activities. Subsequently, these initially generated activity models are extended through incorporating data produced by various users performing these activities in various ways, thus, complete and personalised activity models are learnt. Nevertheless, due to the identified limitations of knowledge-driven approaches, and particularly the identified lack of available expert knowledge, this research focuses on data-driven approaches to HAR.

2.3 Application Domains

HAR is a fundamental component to a broad range of application areas including smart homes and AAL, connected health and pervasive computing [54]. It is commonly used in rehabilitation systems for monitoring the activities of elderly residents to support the management, and also the prevention, of chronic disease. In relation to promoting physical activity, HAR is applied in rehabilitation centers that focus on stroke rehabilitation and those with motor disabilities [6]. Further to this, another common application area is HAR within smart environments, as a key motivation behind HAR research is to monitor the health of smart home inhabitants by tracking their daily activities.

Smart environments are defined in [58] as those that can “acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment”. These environments are an application of

ubiquitous computing that rely on sensor data to perceive the environment, reasoning through data processing to assess how the environment could be changed, and actuators to make changes if required, presented in Figure 2.1. Several factors pertaining to the imperative development of AAL technologies have emerged and continue to rise, such as the cost of healthcare, the ageing population, and the need to support ‘ageing in place’ [59]. AAL is concerned with the provision of remote services and intelligent products to enhance wellbeing and enable independent living for disabled and elderly people through increasing their autonomy and assisting them in carrying out ADLs [12]. The benefits of AAL include increasing quality of life, extending the duration of time people can reside at home by increasing their independence, and providing support for self-care and self-management. Within this domain, numerous smart home projects have been established to promote AAL for the elderly and disabled, for example CASAS [60], Aware Home [61], Gator Tech [62], DOMUS [63], and MavHome [64].

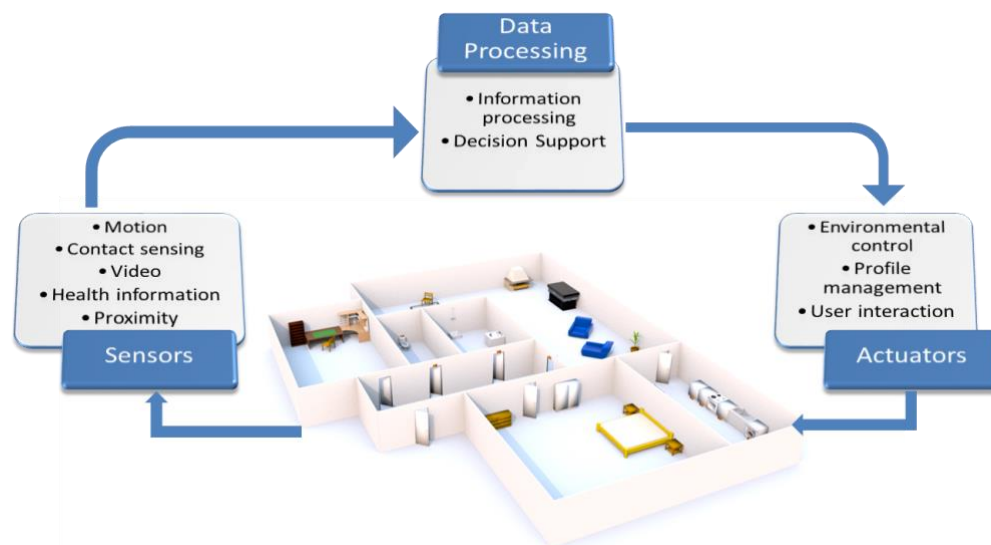


Figure 2.1. Smart environment which involves sensors, data processing and actuators [65]

CASAS is a non-invasive environment developed at Washington State University that is used to analyse the daily activities of inhabitants. This is achieved through the deployment of sensors and the implementation of machine learning and data mining techniques for pattern discovery [7]. Sensors deployed within dedicated CASAS smart environments include motion, temperature, light and door sensors

[66]. An abundance of smart environment datasets generated by CASAS have been made publicly available for research communities to utilise [66]. The Aware Home located within Georgia Institute of Technology is aimed toward assisting the elderly in carrying out ADLs to promote ageing in place. A range of sensors have been deployed for this project, including the use of assistive robots and smart floors to support research, whilst also delivering an authentic domestic atmosphere [67]. Gator Tech was designed to create assistive environments to promote the safety and comfort of inhabitants through the integration of various smart devices [54]. These devices include a smart bed that monitors sleep patterns, a smart bathroom which includes a temperature regulated shower to prevent scalding, smart kitchen appliances and a cognitive assistant to assist inhabitants in performing tasks or to remind them to take their medication, for example [68]. The DOMUS project which operates in the Tuscany region of Italy provides an environment where inhabitants can assess a range of integrated assistive devices [63]. This environment contains in excess of 150 sensors, information providers such as data processors, and actuators [69]. MavHome is a smart environment project deployed at the University of Texas which aims to create an environment simulating an intelligent agent and promotes the comfort of occupants [64]. To achieve this, the environment must be capable of making predictions, reasoning and adapting to the inhabitants [64].

These environments all employ a large number of sensors that capture activity data from a range of sensor modalities. They possess the common aim of supporting smart home inhabitants in carrying out ADLs and providing them with non-intrusive environments to promote their independence and quality of life.

2.4 HAR Process

A number of fundamental stages exist within the data-driven HAR process as presented in Figure 2.2. These include data acquisition through sensors, preparation of the raw sensor data through pre-processing, data segmentation, feature extraction and selection, and finally, classification.

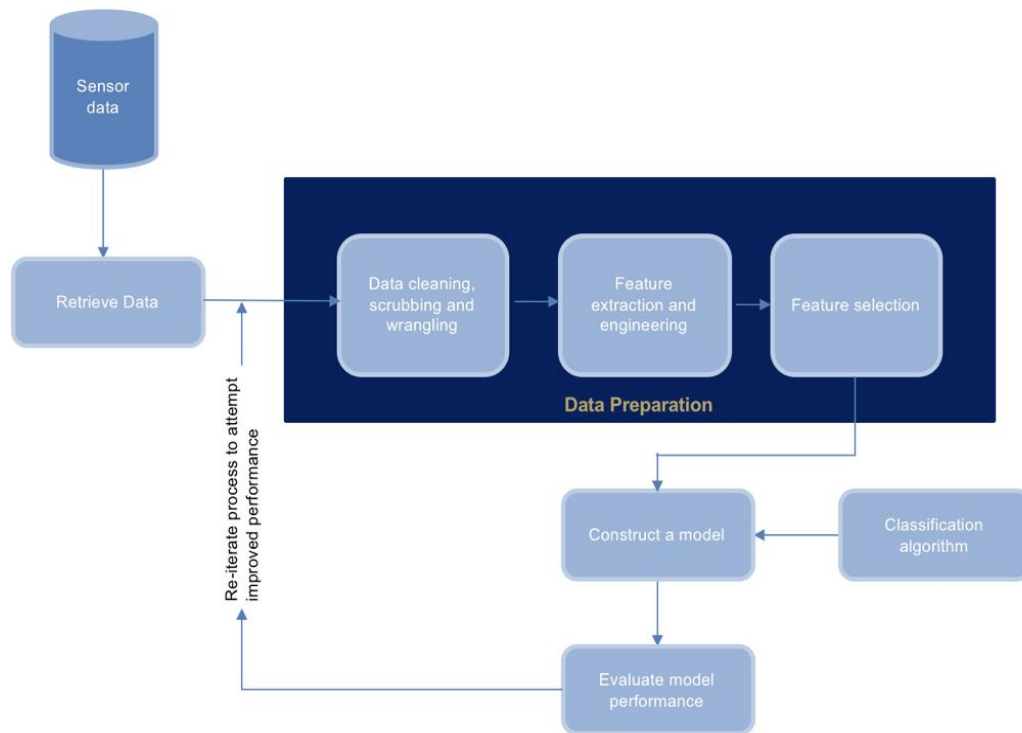


Figure 2.2. The HAR process which includes data acquisition, pre-processing, data segmentation, feature extraction and selection, and classification, adapted from [17]

Section 2.4.1 discusses data acquisition from sensors, Section 2.4.2 discusses data pre-processing which involves identifying types of noise that may emerge and noise handling mechanisms, and Section 2.4.3 outlines data segmentation approaches. Following this, feature extraction and selection are discussed in Section 2.4.4, and finally, Section 2.4.5 considers various classification algorithms.

2.4.1 Data Acquisition

During the data acquisition stage, raw data is acquired from sensors located on the body or placed in the environment [70]. Body-worn sensors typically include accelerometers, gyroscopes, and magnetometers [6]. Recently HAR via signals collected from smartphones have attracted the attention of many researchers [71] due to their ubiquitous nature and as they usually contain embedded sensors such as accelerometers, gyroscopes and magnetometers [72], [73].

An accelerometer is a small electromechanical device used to measure acceleration through responding to vibrations pertaining to movement which are calculated as the alteration in velocity over time [74]. Both static and dynamic

acceleration forces can be measured, for example, a static force depicts the continuous force of gravity, whereas dynamic forces sense vibrations or movements. Triaxial accelerometers are most commonly deployed, where acceleration is measured upon three axis, X, Y, and Z. Figure 2.3 presents an example of the signals produced through performing various static and dynamic activities with a triaxial accelerometer adorned upon the wrist. The static activities within this example include standing, sleeping, and watching TV, whereas the dynamic activities include multiple stand-to-sit and sit-to-stand transitions, multiple stand-to-walk and walk-to-stand transitions, multiple lie-to-sit and sit-to-lie transitions, walking, running, and sweeping.

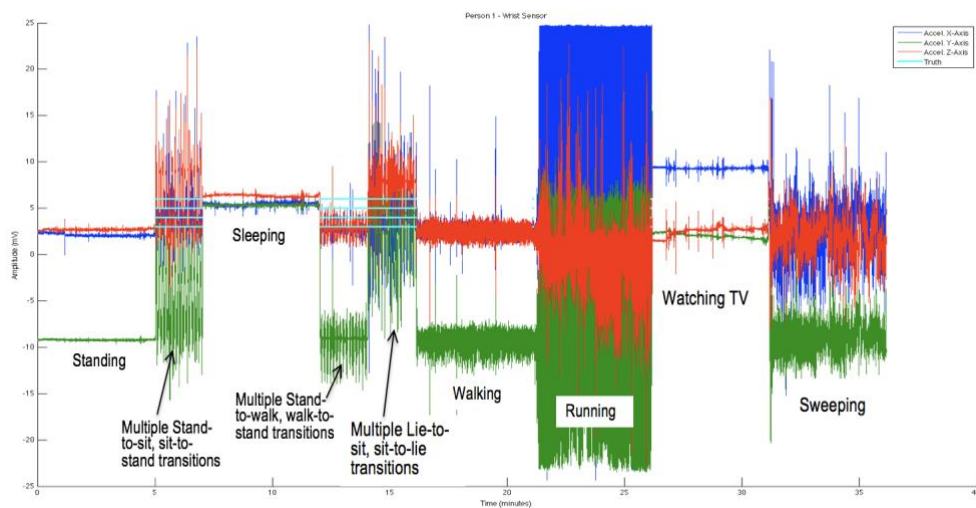


Figure 2.3. Accelerometer signals produced through performing static and dynamic activities

In [75], body-worn inertial sensors were utilised to collect data for the classification of 12 daily activities performed by 6 participants, which included a combination of static and dynamic activities such as walking, running, lying down, standing, sitting, and ascending stairs. The performance of seven classifiers were compared, with the best results achieved by the kNN classifier, followed by the Random Forest and Support Vector Machine classifiers. Furthermore, in a recent study conducted by [52], data produced by a waist-mounted smartphone with an embedded accelerometer was utilised to glean activity data for classification of 12 commonly performed activities. These activities included, however, were not limited

to, standing, sitting, walking, ascending stairs, and combined activity transitions such as stand-to-sit. This study included a wide range of participants (30) with ages ranging from 19-48 to introduce a large degree of variability in the data acquired. Experimental results achieved a classification performance of 96.81%, demonstrating the effectiveness of utilising body-worn sensors for the recognition of the aforementioned activities.

Alternatively, environmental sensors are those attached to objects in the environment rather than the individual performing the activity [76]–[78]. These typically include Passive Infrared (PIR) sensors, contact switches, temperature/light/humidity, vibration, pressure and Radio-Frequency Identification (RFID) sensors [74]. Figure 2.4 presents an example of environmental sensors deployed within a smart environment, consisting of contact switches, motion detectors, and pressure sensors.

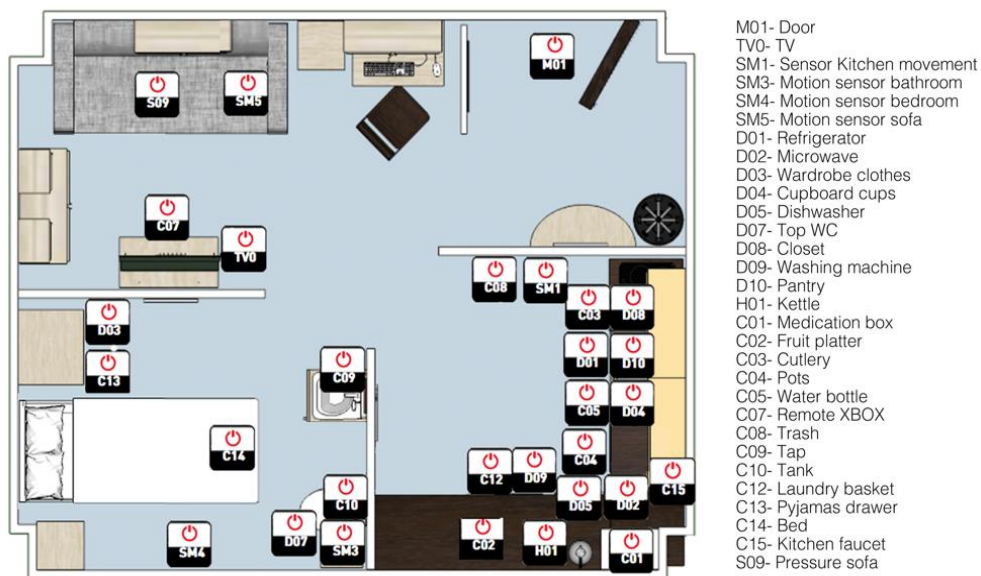


Figure 2.4. Environmentally deployed sensors, including contact switches, motion detectors and pressure sensors [79]

In a study conducted by [76], a range of binary sensors were deployed, comprising both PIR and door sensors, to recognise 4 ADLs. These included meal preparation, eating, relaxing and transitioning from the bed to toilet. Data was streamed over a vast period of 21 months. The large quantity of collected data proved

beneficial for training the proposed classifier, however, it was recognised that a greater number of classes could have been explored. Experimental results achieved 99.36% accuracy when evaluated with a Deep Convolutional Neural Network (DCNN). Another study [77] incorporated data gleaned via binary sensors in the form of motion detectors, contact switches, pressure sensors and float sensors to recognise various ADLs within a smart apartment setting.

It has been recognised that data collection is becoming a vital concern amongst the vast range of challenges within the HAR domain [80]. It has also been recognised that a shortage of large, high-quality HAR datasets exist [13]. According to [81], investigations into generating such refined datasets are required to support a greater availability of data. Table 2.1 presents identified HAR datasets collected through body-worn and environmental sensors, nevertheless, [81] has stated the quality of publicly available datasets is often unclear.

Table 2.1. Identified HAR datasets collected through body-worn and environmental sensors

Data Source	Acquisition	Activities	No. of participants
Opportunity	Body-worn, object and ambient sensors	ADLs	12
Ulster PCH	Body-worn	18 ADLs	141
PAMAP2	Body-worn	18 physical activities	9
WISDM	Body-worn smartphone	6 physical activities	29
UCI HAR	Body-worn smartphone	6 physical activities	30
HASC	Body-worn	6 physical activities	116
CASAS - Aruba	Environmental sensors	11 ADLs	1
SICA	Body-worn	26 ADLs	4
Skoda	Body-worn	10 gestures	1
PSRG	Body-worn smartphone	6 physical activities	4
UCAmI Cup	Environmental sensors	24 ADLs	1

2.4.2 Pre-processing

During the data pre-processing stage, raw data is prepared by handling missing values, reducing noise and data redundancy, and aggregating and normalising the data [7]. Noise may be introduced by the users and sensors during data acquisition which can adversely affect the performance of data-driven techniques, therefore reducing/removing noise is important [17]. Two types of noise exist in the realms of supervised learning, namely class noise and attribute noise [18], described in Section 2.4.2.1 and Section 2.4.2.2, respectively.

2.4.2.1 Class-level Noise

Noise presented at a class level may be introduced during class labelling and annotation. For example, an incorrect label could be assigned to an instance of data, which is commonly known as a labelling error within HAR data acquisition. The prevalent occurrences of class noise appear as either contradictory instances, or misclassification [19]. Contradictory instances emerge when the dataset contains some instances that are the same and occur more than once in the dataset, however they have been assigned different class labels, whereas misclassification noise may emerge when the dataset contains some instances belonging to different classes that have similar characteristics, and consequently are mislabeled as belonging to the same class [82]. In a recent study conducted by [19], class noise materialising through misclassification was examined with specific focus on assessing the robustness of classifiers against this nature of noise. A robustness metric was utilised during investigations, namely Equalised Loss of Accuracy (ELA) which considers both a robustness factor and an initial accuracy factor together, to ascertain the level of robustness of each classifier when evaluated on 10 noisy datasets. Additionally, the quantity of class noise varied between 0-20% per dataset. Experimental results were compared across the suite of classifiers utilised, with the SVM classifier proving the most resilient to noise in comparison to the kNN and Decision Tree algorithms. Further to this, in a recent study conducted by [82], the influence of class noise was evaluated in terms of assessing classification performance with models trained upon low quality, mislabeled class data. Experiments involved injecting twenty levels of class noise, ranging from 5% to 100% and incrementing by 5% per level, into various

authentic and synthetic datasets and two classifiers, namely, SVM and Random Forest. Results demonstrated that as the level of class noise grows, classification performance decreases further. Conclusions stated that the Random Forest model was more robust to class noise in comparison to the SVM classifier, whilst also stating this may have been due to the superior generalisation ability of the Random Forest classifier and its lower likelihood of overfitting in comparison to other models. Furthermore, the number of classes within the evaluated datasets had a substantial influence on performance, as those with lower numbers of classes were more affected by noise than those with several classes, for example, the fewer classes, the quicker model performance decreased, as noise levels had risen. In a recently conducted HAR study [83], the effects of class label noise were explored with activity data collected through a smartphone device. An automatic class annotation method was proposed to eliminate the large time consumption required in manually assigning class labels to HAR datasets, and the effects of noisy class data were evaluated on various supervised classifiers to ascertain their robustness to noise. Both the automatic and manual annotation of HAR data were compared to determine the effectiveness of their automatic labelling method, with results demonstrating a promising 80-85% precision rate. Nevertheless, the adverse effects of class noise were also demonstrated, as the presence of noise could lead to a decrease in f-score of up to 64-74% particularly when evaluated on SVM and Nearest Centroid approaches.

2.4.2.2 Attribute-level Noise

Noise presented at an attribute level may be introduced during data collection where one or more representative attribute values become corrupt or erroneous during data generation, storage, reading, transmission, or processing [18]. Additionally, attribute noise may materialise as missing or incomplete values. For example, sensor failure could result in incomplete or missing data streams being generated and stored. According to [19], attribute noise is more challenging to handle than class noise, and more effort in the machine learning research community needs to be conducted to improve the methods and techniques used in relation to this problem, as very few methods exist in handling attribute-level noise due to its high complexity [18]. A recent study conducted by [84] aimed to examine the effects of

attribute noise on classification accuracy, whilst also improving performance through the application of a simplistic noise reduction algorithm. Various levels of arbitrary attribute noise were introduced to the datasets. Data was first binned per attribute, and subsequently data within each bin was converted into their related z-scores. Data skewness of each attribute was then measured to provide information pertaining to data distributions, and was subsequently transformed through removing attribute noise. Conclusions of this study stated that a negative influence of attribute noise evidently exists, as classification performance was consequently hindered. The effectiveness of the proposed noise handling method was demonstrated as classification performance had improved following its implementation. Additionally, considering the classification models evaluated, the Random Forest classifier had proven most resilient to attribute noise.

Through reviewing the literature, it was found that pre-processing data is an important aspect of the activity recognition process as data-driven approaches rely on good quality data for optimal activity classification. Additionally, according to [18] exploration into data quality, noise detection, and handling mechanisms have enormous scope for investigation to ensure researchers perform accurate data handling.

2.4.3 Data Segmentation

During the data segmentation stage of the HAR process, sensor data is divided into smaller segments to identify the portions of data that are most likely to contain relevant information pertaining to activities being performed [70]. Data segmentation is a challenging task as the beginning and end points of an activity are often difficult to clearly define, and due to human nature multi-tasking and consecutively performed activities regularly occur [70]. Additionally, certain activities may be performed in an interleaved manner. For example, someone could begin a cooking activity, pause to answer a telephone, then continue with the cooking activity. Various segmentation approaches have been established, such as time-based windowing and event-based windowing.

Time-based windowing involves dividing the entire dataset into equal time segments that each comprise a fixed quantity of data per window [74]. An identified

challenge with this approach exists in ascertaining an optimal window length, as a window length that is too small may not include enough representative data to adequately define an activity, whereas a window length that is too large may include an abundance of data spanning more than one activity, thus resulting in a misrepresentative activity window. Consequently, time-based windowing is often preferred in scenarios where a constant sampling rate is employed, for example within accelerometer based HAR [74]. Contrarily, event-based windowing involves dividing the dataset into windows that are comprised of an equal number of sensor events [9]. This technique is often utilised within smart environment research in which environmental sensors are deployed, for example binary state-change sensors in the form of motion detectors or contact switches. The occurrence of high and low event-driven periods cause issues with this technique, as there may be too few or too many occurring interactions with sensors to adequately represent activities [9]. Furthermore, a substantial time lag may occur within a single window, for example during a sleep activity very few sensor activations occur, thus this activity may become embedded within an event window dominated by another activity, and may therefore become misrepresented and overlooked during classification [74]. Figure 2.5 presents a binary data snippet demonstrating both time and event-based windowing, where the time-based window of 15 seconds incorporated 3 sensor events, whereas the event-based window of 15 events incorporated 15 sensor events, regardless of the time period.

TIMESTAMP	OBJECT	STATE	ACTIVITY
2017/10/31 13:32:42.0	SM4	No movement	Brush teeth BEGIN
2017/10/31 13:32:45.0	SM4	Movement	
2017/10/31 13:32:55.0	SM3	No movement	
2017/10/31 13:33:00.0	SM4	No movement	15 second time-based window
2017/10/31 13:33:29.0	SM4	Movement	
2017/10/31 13:33:32.0	SM3	Movement	
2017/10/31 13:33:40.0	SM4	Movement	
2017/10/31 13:33:40.0	SM4	No movement	
2017/10/31 13:33:40.0	SM3	No movement	
2017/10/31 13:33:50.0	SM3	Movement	
2017/10/31 13:33:50.0	C09	Open	
2017/10/31 13:33:50.0	SM4	No movement	
2017/10/31 13:33:52.0	SM4	Movement	
2017/10/31 13:34:01.0	C09	Close	
2017/10/31 13:34:02.0	C09	Close	
2017/10/31 13:34:07.0	SM4	No movement	
2017/10/31 13:34:08.0	SM4	Movement	
			Brush teeth END

Figure 2.5. Examples of time and event-based windowing on binary data, where a 15 second time window is displayed, along with an event-based window comprising 15 events, adapted from [85]

In a study conducted by [9] various approaches to data segmentation were investigated including exploration into both time and event-based techniques. A baseline sliding time window method (fixed length) was compared to various approaches based upon a pre-defined number of sensor events per window. These techniques were evaluated upon 3 smart apartment datasets comprising a number of binary sensors, and classification of activities involved the implementation of an SVM model. Experimental results demonstrated that classification performance peaked when 10-20 sensor events per window were defined, thus also outperforming the baseline time windowing approach. In another study recently conducted [85], time-based windowing outperformed the explored event-based technique when evaluated on a smart home, binary sensor-based dataset. The time period per window was set to 15 seconds, whereas the number of events per window was set to 20. Furthermore, in an attempt to diminish the recognised limitations existing with both time and event-based windowing, dynamic windowing techniques have been emerging in recent years where predefined thresholds and rules may influence the adaptive window size and adjust it accordingly to capture data specific to an activity [74], [85].

Based upon the literature it was acknowledged that the selection and utilisation of segmentation techniques vary depending on the nature of the sensor data gleaned to represent activities. For example, considering the classification of physical activities using wearable sensors with a constant sampling rate, time-based windowing is utilised, whereas considering ADL recognition in smart environments, both time-based and event-based windowing may occur.

2.4.4 Feature Extraction and Selection

Feature extraction may occur in both the time and frequency domains [71]. Due to their high interclass variability, simplistic nature and substantial performance across a range of HAR problems, statistical features are commonly used by HAR researchers in the time domain for the classification of motion signals [71]. These may include calculating the mean, mode, median, variance and standard deviation, as with previous efforts made by [25], [86]. Further to this, frequency domain features such as spectral energy and spectral entropy can be used to provide different perspectives of these signals, and commonly, a combination of both time and

frequency features are extracted [74]. Feature extraction in the time domain is relatively more popular as it requires less computational cost than features extracted in the frequency domain [71].

In a recent study conducted by [87], multi-level feature learning was proposed. The implemented framework consisted of 3 phases of information gathering from data gleaned through a wearable sensor. Phase 1 involved signal analysis to extract low-level features, for example, those extracted from the time and frequency domains as low-level features are prevalent due to their simplistic nature and their adequate performance achieved for HAR tasks. Phase 2 involved the extraction of mid-level features to derive structural signal information as these were deemed more discriminative in representing intricate activities. According to [87], mid-level features describe those that learn the composition of the action. Thus, the Bag of Words (BOW) dictionary learning technique was implemented at this stage to produce mid-level features. Finally, Phase 3 involved the extraction of high-level features to derive semantic information through applying Max-margin Latent Pattern Learning (MLPL). Experimental results had proven the effectiveness of the proposed feature learning framework as state-of-the-art performance was attained on 3 well-established HAR datasets, namely Opportunity [88], Skoda [89] and WISDM [90].

Feature selection involves identifying an optimal subset of discriminative features that can most effectively distinguish activities during classification. A feature vector comprising of those with a high discriminative ability is important as an optimally selected set of features may prove beneficial in diminishing the effects of identified HAR challenges, namely intraclass variability and interclass similarity [87]. Furthermore, the detection and removal of redundant features may enhance classification performance, whilst also decreasing unnecessary computational demands and data complexity [20]. Common feature selection techniques include filters and wrappers, with numerous previously conducted studies stating the implementation of wrapper methods for feature selection outperformed filtering techniques during their experimentation [21]–[25]. Filtering techniques assign a rank to each of the extracted features and are independent of the chosen classifier, whereas wrapper techniques evaluate various feature subsets to discover the optimal feature set specific to the chosen classifier [20], [23], [91]. Further to this, hybrid feature selection approaches have also been proposed more recently and have demonstrated promising results [26]–[29].

2.4.4.1 Filter

Filtering approaches rank the extracted features in terms of their discriminative power and relevance based on statistical criteria, independent of the classifier employed [20], [23], [91]. The features with the highest scores are retained whilst the remaining irrelevant features are discarded. Benefits of this approach include simplicity and time consumption as each filter method is only applied once to the dataset. Nevertheless, an identified limitation of these methods are that feature dependencies are not considered as each feature in the dataset is evaluated separately [20]. Common filtering methods include Relief-F, Information Gain, and Correlation Feature Selection (CFS) [86], [92]. Figure 2.6 presents each step in the filtering process, where the full feature vector is evaluated through applying a chosen filter method, and an optimal subset is subsequently defined following the removal of redundant features. This subset is then applied to the chosen classifiers, and finally the performance of each model is measured.

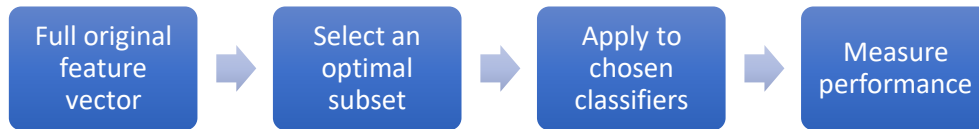


Figure 2.6. The Filter Process [20]

In [92] an evaluation of three filtering methods was presented, namely Relief-F, Gain Ratio and Information Gain. A NN was used to classify the selected input features following the application of each method, where the 20 highest ranking features were retained in each case, representing around a third of the original features extracted in the full dataset. Results of this study revealed the Information Gain filter performed significantly better than the Relief-F and Gain Ratio techniques.

In a recent study conducted by [93], a large quantity of filter methods were analysed in terms of their operation, computational efficiency and predictive quality with regard to classification accuracy. The predominant aim of this study was to provide application recommendations to ascertain which filters generally perform better than others through conducting experiments on 16 datasets generated from various domains. Experiments involved evaluating the feature subsets produced by each filter method on 3 well-established classifiers, namely kNN, LR, and SVM. Conclusions stated that whilst no filter or group of similar filters consistently or

unanimously outperformed the other methods, there were certain filters that performed well on a substantial quantity of the datasets. Thus, recommendations were made based upon this finding. The best performing filters across the collection of datasets included Information Gain and Permutation methods, however, it was also stated that the effectiveness of the filter methods implemented largely depended on the dataset.

2.4.4.2 Wrapper

Wrapper methods determine the most suitable features by evaluating various subsets [23]. These methods search for the subsets most relevant to a specified classifier in order to improve classification performance whilst also considering feature dependencies [20]. The computation time required to execute wrapper methods is, however, large as each identified feature subset needs to be classified to discover which set provides the best classification accuracy [25], and only simplistic classification algorithms may be utilised effectively due to the computational complexity surrounding wrapper methods [94]. Figure 2.7 presents each step involved in the wrapper process, where various subsets of features are derived from the full feature vector and evaluated on the chosen classifier repeatedly until an optimal subset is determined. Following this, the final classification performance is measured.

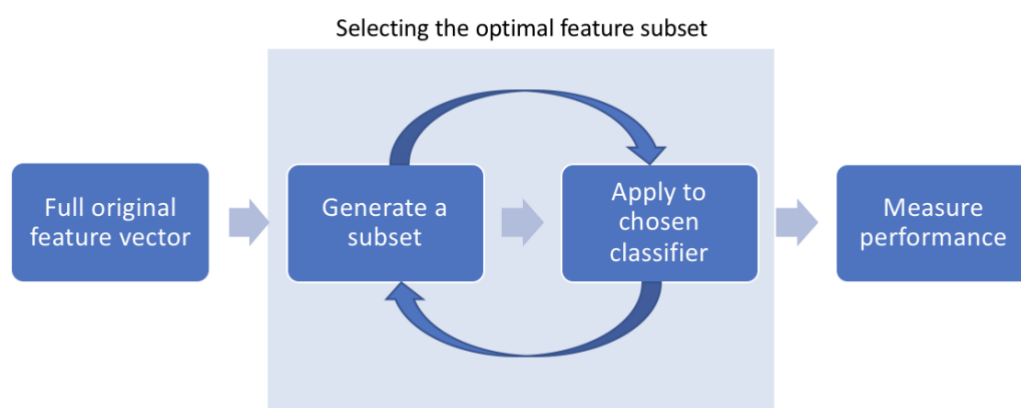


Figure 2.7. The Wrapper Process [20]

Common wrapper approaches include Sequential Forward Selection (SFS), Sequential Backward Selection (SBS) and Single Feature Classification (SFC) [23].

In [22], classification performances following the application of two wrappers (SFC and SFS) and one filtering technique (Relief-F) were compared for a HAR task. The SFS wrapper performed best using the identified top 50 features followed by the SFC and Relief-F methods, although the wrapper techniques required more computation time.

Furthermore, in a study conducted by [95], three wrapper methods were considered, which included SFS, SBS and an evolutionary method. These were each evaluated on ensemble methods, specifically Bagging and AdaBoost, based upon two classification algorithms, namely a DT and NB. Experiments were conducted on 13 multidimensional datasets from various domains. Findings stated that the combination of using the SFS search method along with DT Bagging achieved the best performance in terms of classification accuracy in comparison to other methods. Whilst classification accuracy improved with the ensemble wrapper methods, it was stated that the ensembles required considerably more time for computation in comparison to single classifier methods.

In a recent comparison study conducted by [96], filters and wrappers were assessed and subsequently evaluated with an SVM classifier. The well-established filters utilised in this study included Information Gain, Principle Component Analysis (PCA) and Correlation, whereas the wrapper methods included Genetic Algorithm (GA), Artificial Bee Colony (ABC), and Particle Swarm Optimization (PSO). These methods were compared in terms of their classification performance on a multidimensional dataset, with experimental results demonstrating higher classification accuracies following the removal of redundant features using the wrapper approaches, in comparison to the filter methods.

In a study conducted by [25], a NB wrapper method was compared to eight filters to ascertain which approach was better for selecting optimal feature subsets. These extensively include CFS, Chi Square, Info Gain, Fisher Score, Fast Correlation-Based Filter (FCBF), T-Test, Minimum Redundancy Maximum Relevance and Kruskal-Wallis filters. A HAR dataset was used in this study which contained accelerometer data representing 8 activities. Initially 50 features were extracted from both the time and frequency domains, then subsets were formed following the application of each technique. A NN was used for classification with results showing features selected via the wrapper method outperformed all filtering techniques.

2.4.4.3 Hybrid

According to [26], less effort has yet been explored with hybrid approaches to feature selection in comparison to the common filter and wrapper methods. However, more recently, hybrid approaches involving the combination of well-known feature selection methods, have been emerging in various research domains due to their perceived benefits. For example, hybrid filter-wrapper methods aim to exploit the advantages of both techniques through developing a method achieving enhanced classification performance whilst reducing the computational time required [27].

A study conducted by [27] proposed a hybrid filter-wrapper method to select an optimal subset of features for their chosen high-dimensional dataset, stating the primary aim of their approach was to enhance classification accuracy, whilst also improving robustness. The first stage of their hybrid method involved filtering the entire dataset with Pearson's Correlation to remove redundant features. Subsequently, the next stage of their hybrid method involved the implementation of a wrapper with the SBS search technique along with a Bayesian network to evaluate subsets of the remaining features. Other feature selection techniques, namely Information Gain, Gain Ratio, and Relief-F were evaluated to compare against the hybrid method, with experimental results demonstrating the effectiveness of the proposed method against all others in terms of classification accuracy.

Further to this, [28] recently explored a hybrid filter-wrapper technique based on an Information Gain filter followed by a wrapper employing the SFS search method. The wrapper method involved ensemble learning based upon two classifiers, namely DT and NB models, with which Bagging and Boosting techniques were explored. The approach was evaluated on two multi-dimensional medical datasets and compared against an individual form of the Information Gain filter, and also an individual SFS wrapper. Experimental results demonstrated the benefits of hybrid feature selection as these outperformed the singular filter and wrapper methods. It was also stated that the hybrid methods not only improved classification accuracy, however, also reduced computational costs.

Hybrid techniques are not limited to filter-wrapper approaches, and may involve combinations of similar feature selection methods, for example, two filter methods may be combined to exploit each of their benefits. Recently, [29] conducted

a study in which two filter methods were combined to reduce the dimensionality of the data through the removal of irrelevant or redundant features, whilst also maintaining or improving classification performance. The two filters chosen for experimentation were Information Gain and Chi-Square with the anticipated outcome of combining the derived scores from each method to develop an improved feature selection metric. The proposed hybrid-filter approach was evaluated with two classifiers, namely a C4.5 tree and JRIP rule-based algorithm. Subsequently, the proposed hybrid-filter approach of Information Gain combined with Chi-Squared was compared to their individual filter counterparts. The hybrid-filter method outperformed the singular filter approaches in terms of achieving a higher classification accuracy with the resulting feature subset, proving the benefits of combining these techniques.

Based upon the literature, it is apparent that feature selection is a fundamental stage within the HAR process. Previous efforts have been explored most commonly with time and frequency domain features derived through signals produced by wearable sensors, however relatively less effort has been recognised in exploring feature selection pertaining to environmentally deployed binary sensors. Additionally, the exploration of hybrid feature selection has been emerging more recently, thus the opportunity for further exploration exists.

2.4.5 Classification

Activity classification and performance evaluation is the final stage of the HAR process, within which the set of features extracted in the previous HAR stage are used as inputs to a classification model. With supervised learning, a training phase is required where the classification algorithm is presented with feature vectors along with their corresponding class labels available as ground truth [70]. Once a model is trained, the classification step utilises this trained model to map newly presented feature vectors to a set of predicted class labels [70]. Data driven classification algorithms can be generally deemed as either generative or discriminative models [54]. Generative models aim to produce a complete representation of the input space, commonly through the use of probabilistic models such as the simplistic Naïve Bayes (NB) algorithm [55]. Discriminative models map

representative input data to their outputs, commonly through the implementation of rule-based algorithms such as Decision Trees (DT), or through Neural Networks (NN) and Support Vector Machines (SVM), for example [54]. Table 2.2 presents comparisons generated between each of the considered classifiers, which are discussed further in the following Section.

Table 2.2. Data-driven classification algorithms

	Model	Description	Advantages	Disadvantages
Generative Models	Naïve Bayes (NB)	Probabilistic model based upon Bayes theorem that involve calculating posterior probabilities within activity classes [75]	Easy to build and implement as these models do not require intricate iterative parameter estimations [97]	These models do not possess the ability to model temporal data explicitly
	Hidden Markov Model (HMM)	Creates hidden states from data observations and aims to discover relationships between these states and their corresponding observations [98]	Highly capable of modelling temporal data dependencies found in simple activity sequences [70]	Can find more complex activities difficult to recognise, for example ADLs [8], and also concurrent or interleaved activities [8]
Discriminative Models	Support Vector Machine (SVM)	Based on statistical learning theory [74] Attempts to increase accuracy and robustness, whilst avoiding the problem of overfitting [74]	Highly accurate Highly robust	Poor kernel choice may affect optimal SVM configuration [99] and models can experience confusion when classifying similar activities [100]
	Decision Trees (DT)	Favored for inductive inference [74] Presented in a tree structure where each node (branch) represents an input feature, and each leaf represents a class label [71]	Their low complexity and non-intricate implementation are beneficial for HAR tasks [17]	Can experience difficulty in handling non-linear relationships [17]
	k-Nearest Neighbour (kNN)	Very simplistic classifier that is known as a “lazy learner” as no training stage is executed, and therefore, no model is pre-built [101]	Highly accurate and robust [102]	Model performance largely depends upon the distance measure applied [102] and requires high computational costs [103]
	Neural Networks (NN)	Connectionist models that map inputs to their corresponding outputs. NNs are based on layers of nodes (neurons) that are connected to one another, with weights and bias determined for each connection	NNs are capable of modelling complex, non-linear relationships which is valuable for application in the HAR domain [47]	Require high computational costs during training

2.5 Classification Algorithms

This Section provides further, comprehensive descriptions of the classification algorithms previously introduced in Table 2.2. The generative models, namely NB and HMM, are discussed in Section 2.5.1. Contrarily, the discriminative models, namely SVM, DT, kNN, and NN are described in Section 2.5.2. Finally, Section 2.5.3 discusses ensemble methods for classification.

2.5.1 Generative Models

The NB classifier is a generative approach that has produced encouraging results for HAR [9]. This simplistic model is based upon Bayes theorem and involves calculating posterior probabilities within activity classes, therefore during the testing stage using unseen data, the class with the highest probability is considered most likely to be correctly classified [75]. This classifier makes assumptions of the independence and normality of the input features [104]. According to [101], NB classifiers are easy to implement and are computationally efficient, though classification performance may be hindered if too few data instances exist. This classifier was implemented within HAR studies in [38]–[40]. In a study conducted by [40], temporal patterns of actions were identified to subsequently recognise a series of actions that represent full activities. A HAR dataset, namely Opportunity, was evaluated within which 4 participants performed 5 activities, including, relaxing, eating, and cleaning. For classification of these activities, 3 well-established classifiers were implemented and compared, namely a NB, an SVM, and a kNN. Experimental results demonstrated the effectiveness of the NB model which outperformed all other classifiers, achieving an accuracy of 98.0%. Further to this, in [105] various classifiers, such as a NB, DT, and Random Forest, were implemented to recognise 6 activities gleaned through a smartphone-embedded accelerometer. These activities included walking, ascending stairs, descending stairs, standing still, lying down and sitting. Experimental results showed the NB model was most efficient in terms of time taken to construct the model, however the Random Forest model was most effective in terms of classification accuracy achieved.

Hidden Markov models (HMM) are another example of generative probabilistic approaches commonly used for HAR tasks [70]. HMMs create hidden states from data observations and aim to discover relationships between these states and their corresponding observations [98]. Two parameters, namely, the State Transition Probability and State Emission Probability are considered to achieve this [98]. HMM classifiers are very capable of modelling temporal data dependencies found in simple activity sequences [70], however, they can find more complex activities difficult to recognise, for example recognising ADLs [8]. HMMs have been used in [98], [106]–[108] for HAR tasks. In [98] 5 participants were tasked to perform various activities with 4 body-worn accelerometers attached to both ankles, their belt, and their chest. The activities included those of a static nature, such as standing and sitting, as well as transitional activities, such as transitioning from a sitting position to standing still. An HMM was implemented to recognise these activities, where HAR performance was evaluated in terms of accuracy achieved. Results obtained demonstrated superiority of the ankle-worn sensors in comparison to the belt and chest locations, where the left and right ankle locations achieved 59.52% and 55.92%, respectively. Further to this, in a recent study [106] a two-phase HAR framework using hierarchical HMMs was proposed in an AAL setting. Within Phase 1, specifically the “Detection” phase, online data streams were segmented and real-time HAR was performed per activity instance received through implementing an HMM. Phase 2 was deemed the “Correction” phase in which a Joint Probabilistic Distribution Function was extracted, per class. This was then used to estimate the probability of each activity instance belonging to each class. The HAR framework was evaluated on two non-public datasets, which involved various ADLs such as eating a meal, sleeping, taking medication and personal hygiene. Results demonstrated the effectiveness of the proposed HAR framework, achieving 65.20% accuracy with Dataset 1, and 60.00% accuracy with Dataset 2. Nevertheless, conclusions stated some activities were more difficult to discriminate due to the same sensors being activated, for example, the enter home and leave home activities.

2.5.2 Discriminative Models

A notable discriminative classification algorithm is the SVM. This algorithm is based on statistical learning theory, and attempts to increase accuracy and robustness, whilst avoiding the problem of overfitting [74]. In their most basic form, SVMs are presented as linear binary classifiers, where the algorithm attempts to discover a suitable hyperplane within the feature space to linearly separate data presented from two possible output classes [99]. Adjustments may be made to simplistic SVMs in order to generate predictions for multiclass classification tasks, as many problems, including HAR, typically contain multiple classes [99]. For example in a study conducted by [109], an Error-Correcting Output Code (ECOC) method was implemented to enable multiclass classification through the SVM classifier. Kernel choice is an important consideration when optimising SVMs, as kernels are used to handle the inseparability problem through introducing extra variables [99]. SVMs are increasingly being implemented within HAR studies, such as [30]–[33]. In [33], SVMs were chosen for HAR through body-worn sensor data, in which multiple accelerometer sensors and placements were explored to estimate energy expenditure whilst performing activities of various intensity levels. The stated rationale for classifier choice was based upon promising results achieved in their previous works, within which SVMs were employed with a polynomial kernel. Experimental results indicated that energy expenditure was optimally estimated through the deployment of a single accelerometer positioned in close proximity to the center of mass, for example, located either on the participants chest or waist. Furthermore, in [32] comparisons were made between various SVM configurations aimed at recognising 6 commonly investigated activities of both static or dynamic nature, namely, lying down, sitting, standing, walking, ascending stairs, and descending stairs. The proposed SVM, using a One-vs-One (OVO) method and a linear kernel, was compared to an SVM involving a polynomial kernel, in addition to an SVM using a gaussian kernel. Experimental results produced during this study revealed that the polynomial kernel achieved the best classification performance, followed by the linear kernel when evaluated on wearable sensor data.

DTs are another example of discriminative models that are favored for inductive inference [74]. These classifiers are presented in a tree structure where

each node (branch) represents an input feature and each leaf represents a class label [71]. According to [110], there are three important aspects to consider whilst implementing DTs, namely, splitting, stopping and pruning. Splitting involves dividing parent nodes into child nodes with enhanced purity relating to the target class and based upon the input features. The splitting aspect ceases when a stopping criteria is achieved, which is usually applied to prevent the built model becoming too complex, and thus, overfitting. As an alternative, instead of applying stopping criteria, pruning may be applied to a complex tree to ascertain the optimal size. Pruning involves removing nodes that provide the minimal necessary information, specifically redundant nodes, to reduce the size of the tree, and consequently, reduce its complexity. DTs are commonly used for HAR due to their low complexity and non-intricate implementation, though difficulty has been found in handling non-linear relationships [17]. DTs have been investigated in [34]–[37]. In [36], HAR was explored with wearable sensors to detect 4 common activities, namely, lie, sit, walk and jog, within an IoT scenario. Two classifiers were initially considered, namely DT and NB classifiers, however, through initial experimentation stages, the DT was deemed most suitable in terms of complexity and computational efficiency, thus, further experiments were conducted with this algorithm. The DT model achieved 95.83% accuracy during classification, thus demonstrating its effectiveness. In another study [37], a HAR system was implemented based upon an accelerometer within a smartphone. A total of 12 ADLs were performed by 66 participants, which included, for example, standing, walking, sitting, jogging and ascending stairs. A J48 DT, kNN, Logistic Regression (LR) and an NN were implemented and the HAR performances of each were compared, within which the DT and kNN models proved most effective in classification performance overall.

The kNN classifier is known as one of the longest-established, most simplistic, and accurate models for classification and regression tasks [102]. kNNs are known as “lazy learners” as they do not execute an explicit training stage, instead, the training data is stored and all computation is conducted during the test stage [101]. This classifier is highly robust to noise and no prior knowledge of the data is required for classification, for example, data distribution information [102], though high computational costs are required for utilisation, and the classification performance achieved is largely reliant upon the applied distance measure [102], [103]. kNN models have been implemented for HAR tasks in [41]–[43]. In [41], 5 common

ADLs were classified using a kNN classifier in an AAL scenario. Data was gleaned through a smartphone-embedded accelerometer and gyroscope, where a total of 10 participants performed the 5 activities, namely, sitting down, standing still, walking, ascending stairs, and descending stairs. The effectiveness of the kNN classifier was demonstrated during this study, with experimental results attaining a classification accuracy of 88.00%. Further to this, real-time HAR was explored recently using wearable sensors in [42], within which the performance of 5 classifiers were compared, namely a kNN, NN, SVM, DT and NB. A total of 15 participants were tasked to perform 8 common activities, such as sitting, standing and walking. Results demonstrated the superiority of the kNN model, which achieved the highest classification accuracy of 96.70%, followed by the SVM classifier.

NNs have been attracting attention recently and are becoming a popular classifier for HAR tasks [111]. NNs are discriminative models defined by [112] as a “biologically-inspired programming paradigm which enables a computer to learn from observational data”. The Multi-Layer Perceptron (MLP) is a notable type of NN often used for activity recognition tasks [44]–[46] which consists of an input layer, one or more hidden layers, and an output layer [113]. They are capable of modelling complex, non-linear relationships, which is valuable for application in the HAR domain, and have been established as one of the most effective NN methods for predictive power [114]. In a HAR study conducted by [115], explorations into designing an optimal MLP classifier were conducted to recognise 6 common activities. According to [115], the classification performance achieved by NNs are largely reliant upon selecting the optimal number of neurons in the hidden layer, thus a new hidden neuron selection method based on convergence theorem was proposed to enhance HAR performance. The effectiveness of the proposed formula to define an optimal NN architecture was demonstrated, as experimental results obtained 98.32% accuracy when evaluated on a publicly available UCI HAR dataset.

Due to recent advancements with technology, the computational capacities required by more complex NN architectures can be attained as modern GPU clusters can provide better performance and support [116], and since the emergence and success of deep learning in domains such as natural language processing [117], image recognition [118], and speech recognition [119], these methods are now being applied to HAR problems. Though according to [120] the efficiency of deep architectures in comparison to simple NNs remains unclear. Deep Neural Networks

(DNNs) can be perceived as a traditional NN with many hidden layers [119]. These additional layers enable automatic and complex feature learning combined with the classification phase [121] which is anticipated to be beneficial for HAR problems involving the recognition of intricate ADLs within smart environments, previously attempted by [122], [123].

Two commonly implemented, complex NN architectures are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs are becoming a popular method for HAR as they provide two core advantages for application in time-series HAR scenarios: local dependency and scale invariance [120]. CNNs are capable of preserving feature scale invariance due to the inclusion of a pooling layer, therefore variations of an activity, for example due to different walking styles or paces, can be effectively captured [124]. CNNs are also able to capture local dependencies of activity signals in the convolution layers, which means correlation is likely to occur for signals collected nearby [124]. In a recent study conducted by [76] a Deep Convolutional Neural Network (DCNN) was implemented for the classification of 4 commonly performed ADLs within a home monitoring environment. These activities included meal preparation, eating, relaxing and making a transition from bed to toilet. The DCNN architecture consisted of two convolutional layers each followed by max-pooling layers, and subsequently two fully connected layers. The process involved converting binary sensor data produced by 31 wireless passive infrared (PIR) motion sensors and 4 door sensors, into representative activity images for each of the activities defined. These images were then used to train and test the proposed DCNN classifier which produced an accuracy of 99.36% for ADL recognition. Although results produced are substantial, a larger number of activity classes could be investigated. As for the RNN models, these have proven beneficial in handling sequential input data and exploiting temporal relationships, which is fundamental for successfully recognising human activities [125]. These networks are recurrent in that a cycle is formed between some connections to consider previous time steps whilst computing the current state of the network [126]. In a recent study by [127], RNNs with LSTM are used to recognise activities in groups of participants, for example whilst playing a game of volleyball, with the intention of recognising events and interactions occurring between subjects, and actions performed by individual subjects. The network architecture developed in the study is named “Confidence-Energy Recurrent Network (CERN)” as it

involves a confidence measure and an energy layer. Results show that the CERN architecture achieves better performance than other previous efforts made to recognise group AR via RNNs, which also performs well with uncertainty.

As previously mentioned, the efficiency of deep architectures in comparison to more simplistic NN models remains unclear [120]. In a HAR study conducted by [44], 11 ADLs were classified by both an NN and a DNN, in which the NN outperformed the deep architecture. Conclusions stated the DNN performance may have suffered due to the low amount of training data available, as a large amount of data is required to demonstrate the potential of DNNs. According to findings in [128], the implementation of a simplistic NN architecture comprising two or three layers can be most effective for HAR tasks. Their experiments involved evaluations upon two HAR datasets, namely, WARD and UCI_DB, in which several NN architectures were explored. Experimental results demonstrated the effectiveness of a shallow NN which achieved 99.2% and 99.7% in terms of classification accuracy on the WARD and UCI_DB datasets, respectively. In comparison, the best performing deep NN architecture was that of a CNN model, which achieved 97.7% and 94.2% on the aforementioned datasets, respectively. Conclusions of these investigations recommended the implementation of shallow NN architectures for HAR tasks rather than exploring deeper architectures, particularly in cases where a large amount of data is not available.

2.5.3 Ensemble Methods

Ensemble methods have attracted considerable research attention recently due to their ability of improving classification performance [48]. In [50], it was recognised that the goal of enhancing generalization capabilities is the primary motivation towards exploring ensemble methods. This approach involves combining several base models to generate an ensemble, instead of depending solely on the predictive capability of a single classifier [74]. Ensemble learning includes two main considerations: ensemble generation and ensemble integration [129]. The generation phase involves ascertaining the size of the ensemble and subsequently generating the chosen base models. If the models are generated through implementing a consistent induction algorithm, it is recognised as a homogeneous method, whereas a

heterogeneous ensemble method involves generating base models through implementing diverse algorithms [130]. In [131], an ensemble of random forest classifiers was proposed with the aim of generating a more accurate, stable classifier to recognise activities from the PAMAP physical activity dataset. HAR performance attained was 93.44% accuracy, and the generalization capability of the produced classifier had improved significantly. In [132], multiple HMM base models were combined using a decision template method to recognise activities collected by a smartphone-embedded triaxial accelerometer. Their approach addressed the interclass similarity and intraclass variability HAR challenges, with results showing that the ensemble generated performed significantly well with data representing six activity classes and collected by 30 participants. Further to this, in [133] an anomaly detection method within ensembles was explored, in that faulty base models were identified and removed from the ensemble method to enhance HAR performance. Two methods were explored at a fusion level to identify faulty base models: the first method involved comparing the class decision outputs provided by each base model with the final fused output through Mahalanobis distance, whereas the second method involved evaluating the mutual information between all base models within the ensemble. Results demonstrated the effectiveness of the proposed approach, as the implemented methods revealed performance improvements with both evaluated HAR datasets in comparison to retaining all original base models.

More recently, ensemble methods based upon deep learning techniques have been explored in areas including video/audio based HAR, which have demonstrated promising results. For example, [134]–[136] proposed deep learning ensemble techniques for HAR, which demonstrated enhanced robustness and achieved promising results. In [134] a hybrid deep NN ensemble for vision-based HAR is proposed. Multiple CNN base models, each trained with diversified input data and varying model initialisation parameters, were combined using a fusion function to ascertain the final class output. Results demonstrated very high HAR performance, specifically 99.68%, when evaluating the proposed approach with the UCF50 dataset, thus outperforming all benchmarked comparison studies. Further to this, [135] proposed an ensemble of deep NNs to recognise voice-based activity detection in which the final output decision was determined by fusing the estimated class output produced by each base model through a weighted combination method. The proposed approach outperformed the benchmarked conventional algorithms

evaluated, namely an SVM and a single deep NN, thus demonstrating its effectiveness. In [136] an ensemble comprising of multiple deep LSTM base models was proposed for the purpose of HAR using wearable sensors. The effectiveness of the proposed approach was demonstrated during their experimentation, as the proposed ensemble of deep LSTMs outperformed the benchmarked single deep LSTM model when evaluated on 3 wearable HAR datasets, namely Skoda, Opportunity and PAMAP2. Nevertheless, deep learning methods require a large quantity of training samples to demonstrate their potential and improve classification performance [137].

2.5.3.1 Ensemble Generation

During ensemble generation, data partitioning is a commonly considered approach aimed towards diversifying the input data of the base models so that the subspaces of inputs become complementary [138]. Two commonly implemented data partitioning ensemble methods include Boosting and Bagging, which are implemented to combine several classifiers that have been trained on various diverse subsets of the training data [74]. Boosting involves the combination of multiple base classifiers to generate a strong committee classifier that may provide significantly enhanced performance in comparison to the base classifiers, achieved through reweighting the misclassified data samples and therefore boosting their performance [74]. Adaptations of the well-established AdaBoost method exist, namely RUSBoost and SMOTEBoost, in which random undersampling or SMOTE is applied to the training data of each base classifier, along with the reweighting phase in accordance with the Adaboost algorithm, previously explored in a study conducted by [139]. Both RUSBoost and SMOTEBoost introduce a great level of variability through creating or eliminating data samples, leading to enhanced robustness to noise.

Contrarily, Bagging involves calculating the average of outputs generated by each base classifier, where each model is trained with diverse training sets comprising of data produced through sampling with replacement [129]. Well-established bagging-based techniques to partitioned data include OverBagging, UnderBagging and SMOTEBagging. Specifically, the SMOTEBagging method has been recommended for applications in multi-class imbalanced data problems where

the data samples contained within each bag are considerably diverse [140]. In a recent study, [141], two bagging-based hybrid methods were proposed to deal with imbalanced datasets, namely ADASYNBagging and RSYNBagging. The ADASYNBagging method involved implementing the bagging algorithm as well as the ADASYN-based oversampling method. Contrarily, the RSYNBagging method involved implementing the ADASYN-based oversampling technique as well as random undersampling alongside the bagging algorithm. The classification performances of the proposed hybrid approaches were subsequently compared against UnderBagging and SMOTEBagging techniques and evaluated on twelve datasets, with promising experimental results obtained. The effectiveness of the proposed hybrid methods were demonstrated as they achieved superior performances in comparison to the considered benchmark methods on eight of the twelve datasets evaluated.

Another approach considered during ensemble generation is to manipulate the inputs of the base classifiers at a feature level, for example, training the base models on various diverse feature subsets [138]. A common approach in achieving diverse feature subsets involves implementing the Random Subspace method, in which the full feature vector is decomposed into smaller subsets at random, thus each base model is subsequently trained on a diverse, randomly generated feature subset [142]. In [142], various ensemble generation methods were implemented including Random Subspace, Bagging, Ensembles of Nested Dichotomies (END), Rotation Forest and AdaBoost, which were evaluated with two base models, namely, an SVM and Random Forest, for the purpose of HAR. Six common activities were considered, including walking, ascending stairs, descending stairs, standing, lying down and sitting, which were acquired through an accelerometer-embedded smartphone. Experimental results demonstrated the effectiveness of the Random Subspace method which outperformed all remaining considered methods, achieving 99.22% accuracy. Another study [143], proposed a novel feature grouping technique which utilised Localised Generalisation Error Model (L-GEM) to evaluate the proposed Multiple Classifier System (MCS) for the purpose of HAR. Within the proposed approach, the Genetic Algorithm method was utilised to select random subsets of features. Experimental results demonstrated the effectiveness of the proposed feature grouping technique, achieving an accuracy of 87.35%, which

outperformed the Random Subspace benchmark method, which achieved 83.78% accuracy.

2.5.3.2 Ensemble Integration

During the ensemble integration phase, the output predictions produced by each base model are combined in attempt to enhance classification performance by attaining a single outcome [129]. Several integration strategies exist, which may be explored at either a class label level, a support level, or a trainable level, according to [137]. Integration techniques at a class label level typically involve voting strategies, within which each base classifier may vote for a particular class, thus the final output prediction is determined through employing either a majority voting or weighted majority voting strategy. Majority voting determines the final output prediction based upon the class label that has been selected most frequently, or unanimously, by the base models. Contrarily, weighted majority voting involves assigning weight values to each base model, often based upon their classification performances during training, where the classifier attaining the highest output after weight assignments is awarded the overall class prediction [137]. In a study conducted by [144], majority voting was implemented to ascertain the final outputs of an ensemble approach based on AdaBoost. Three weak classifiers were used, namely a DT, LR, and Linear Discriminant Analysis (LDA). In addition to AdaBoost, Bagging and Stacking methods were also explored, with the best performance produced by the Bagging approach. Another study [131], used weighted majority voting with an ensemble of Random Forest classifiers. Each classifier was assigned different weights per activity, with the final outcome attained through combining the classification outcomes from each base model via the weighted votes.

Integration techniques at a trainable level consider the chosen fusion weights during the learning process and implement optimization strategies to increase classification performance whilst also reducing computation cost [137]. These include weighted summations of hypotheses where higher weights are assigned to those with lower error rates, and the Dempster-Shafer theory to handle uncertainty in the decision-making process. In [145] the output predictions of multiple SVM

models, trained on diversely generated feature subsets, were subsequently integrated using the Dempster-Shafer fusion strategy. The four-step method involved creating decision templates for all training instances, calculating the proximity between decision templates and classifier outputs, computing the belief degrees for each output class, and finally, applying the Dempster rule to integrate the degrees of belief derived from each base classifier.

Finally, the support function integration method involves computing an output decision score for each base classifier, which is derived from the estimated likelihood of a class [138]. This class estimation may be computed as an *a posteriori* probability obtained through probabilistic models, using fuzzy membership functions, or through combining NN outputs according to their performance. In [146], five base models were generated and combined using an average of probabilities fusion method to recognise six activities. This method involved using the average of the probability distributions for each base classifier to make a final class decision, which achieved the best HAR performance in comparison to a majority voting method that was also implemented during this study. Another study [89], explored support function integration in which a Naïve Bayesian fusion method was compared to a majority voting approach to fuse the outputs of multiple HMM base classifiers. The Naïve Bayesian method involved calculating the posterior probabilities of the HMM outputs, which achieved the best HAR performance during the study.

Through reviewing the literature, it has been recognised that many well-established classifiers have been implemented within HAR studies. Particularly, NNs have been attracting more attention recently due to advances in technology enabling the implementation of more complex architectures, in addition to simpler architectures providing benefits in modelling complex, non-linear relationships, which has been deemed valuable for application within the realms of HAR. Further to this, ensemble methods have been attracting considerable research interest due to their ability in improving classification performance, in addition to their perceived benefits in comparison to individual classifiers. Thus, ensemble methods demonstrate scope for further exploration.

2.6 HAR Challenges

Taking into consideration the body of research reviewed, many research challenges associated with HAR have previously been identified due to the complex nature of human activities and available data sources. These challenges include Intraclass Variability, Interclass Similarity, and Class Imbalance. The ability to recognise concurrent and interleaved activities are also challenges associated with HAR.

2.6.1 Intraclass Variability

Intraclass Variability is a challenge introduced during data collection and occurs as the same activity may be performed differently by different individuals [70], and a single individual never performs a particular activity in the same exact way, as the performance of an activity can be influenced by factors such as fatigue or stress [147]. For example, an individual's walking style may vary depending on the time of day. Walking may be more dynamic in the morning following a long rest and less energetic in the evening following a long and stressful day at work, leading to a variation in the individual's walking style [70]. This challenge may be addressed through increasing the quantity of training data and ensuring this data contains as many variations as possible of each activity being performed.

In a study conducted by [87], a 3-phase framework was proposed to exploit multi-level feature learning in an attempt to improve HAR performance on a dataset exhibiting the intraclass variability problem. The Skoda dataset was evaluated, which contains accelerometry data gleaned through movements performed by a single person. The produced signal was interpreted during each phase of the framework to extract representative features of activities performed. In Phase 1, the extraction of low-level features transpired which included those extracted from both time and frequency domains. Subsequently in Phase 2 of the implemented framework, the focus was aimed at learning structural compositions to derive mid-level features, which involved implementing the BOW learning method. Finally, in Phase 3, MLPL was applied to gain semantic interpretation of the signal at a high-level, thus exploiting the intraclass variations of each performed activity. The

proposed framework was evaluated on 3 classifiers, namely, kNN, SVM, and Nearest Centroid Classifier (NCC). Experimental results demonstrated the effectiveness of the MLPL method in recognising activities exhibiting intraclass variability, with results attaining state-of-the-art on the Skoda dataset. Another recent study [148], proposed the implementation of a margin mechanism to handle intraclass variability within 3 HAR datasets, namely, Opportunity, PAMAP2 [149] and UniMiB-SHAR [150]. The effectiveness of the proposed margin mechanism was evaluated with 4 NN classifiers exploiting deep learning. The aim of this was to expose superior discriminative features, thus diminishing the adverse effects of intraclass variations. The resulting method was defined as Margin-based Loss function, as the implemented softmax function was adapted with the margin mechanism. Experimental results demonstrated the effectiveness of the proposed approach, as the performances achieved had outperformed comparative experiments.

2.6.2 Interclass Similarity

Interclass Similarity is a common HAR challenge that occurs when certain activities generate similar sensor characteristics, though they are physically different. For example, the activities of walking upstairs and walking downstairs both produce similar sensor characteristics. Thus, they can be difficult to discriminate between during classification, causing diminished classification performance [70]. Consequently, the interclass similarity problem is deemed an imperative issue in the realms of sensor-based HAR that requires substantial consideration, according to [103].

A study conducted by [151] found that their approach to HAR was successful in discriminating between similar activities. Their investigations considered both moving and stationary activities, namely walking, walking upstairs, walking downstairs, sitting, standing and lying down. Data to represent these activities was collected by 30 participants using a smartphone embedded accelerometer and gyroscope. A CNN was used in this study, with results being compared to other state-of-the-art approaches to classification. Results for classifying the three similar moving activities using the proposed CNN achieved 99.66% accuracy, and results for classifying both moving and stationary activities achieved an accuracy of 94.79%.

Further to this, in [103] a method to deal with the interclass similarity problem was proposed, which involved decomposing the multiclass HAR task into manageable binary classification tasks. The One-Versus-All (OVA) method was implemented in an attempt to enhance discriminative abilities between activities that exhibit similar sensor characteristics. The proposed approach was evaluated upon 2 HAR datasets, namely WISDM and PSRG, and 3 well-established classifiers, namely RF, kNN and DT. Experimental results obtained during this study outperformed those reported within identified benchmark studies in terms of classification accuracies attained, particularly whilst classifying similar activities. Optimal performance was achieved through implementing the OVA decomposition method in conjunction with the RF classifier.

2.6.3 Class Imbalance

Class Imbalance is a term used to describe a large quantity of data that is unevenly distributed [152]. Imbalanced datasets usually contain majority classes that contain a substantial number of instances, and minority classes that contain fewer instances [152]. Majority classes can overwhelm standard classifiers, consequently leading to minority classes being neglected during classification, thus achieving poor performance. The class imbalance problem arises within the HAR domain as some activities occur often or continuously whereas others occur infrequently or periodically [70]. Figure 2.8 presents an example of imbalanced class data within a HAR dataset.

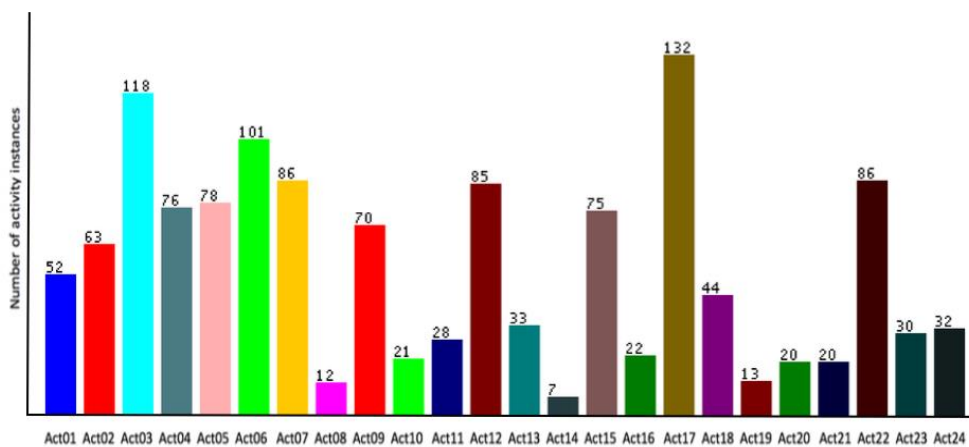


Figure 2.8. Example of class imbalance existing within HAR data produced for the UCAmI Cup [79]

Various techniques have been identified and investigated to address this HAR challenge. Many imbalanced data handling techniques focus on data pre-processing, for example, sampling methods such as undersampling, oversampling and synthetic sampling, whilst other techniques focus on the classification stage and model evaluation [153].

In a recent study conducted by [154], an undersampling technique based upon clustering was implemented, namely Fast-CBUS. This method involved clustering the minority classes of each of the 11 evaluated datasets, and constructing a classifier per cluster, namely an SVM. Following this, majority classes existing within close proximity to each clustered minority class were undersampled to achieve a balance within each clustered model. The classification performance achieved by the proposed method was compared to that of 4 other undersampling techniques, as well as an oversampling approach, namely SMOTE. Experimental results revealed the Fast-CBUS method outperformed all other undersampling techniques, however, the SMOTE oversampling technique achieved better performance than the proposed technique. Nevertheless, a substantial trade-off existed between performance achieved and computational costs. Other recent studies focusing on handling the class imbalance problem include those by [155], [156]. In [155], clustering-based undersampling was proposed. During data pre-processing, clustering was implemented within which the quantity of majority class clusters was to be equivalent to the quantity of minority class instances. Two methods were proposed: the first involved representing the majority class using cluster centers, whereas the second method involved representing the majority class using the cluster centers' nearest neighbours. The second method proved most effective during experimentation, where the proposed approach outperformed 5 benchmarked techniques, involving 3 UnderBagging methods, SMOTEBagging, and RUSBoost. Contrarily, in [156] a two-phase resampling technique is proposed. Within the first phase, OVO is applied to remove noisy data, thus ensuring the resampling method only samples data of higher quality. Subsequently, within phase 2 each minority class was oversampled to balance the distribution of classes across the dataset. During experimentation, various levels of noise were investigated. The proposed method performed reasonably well with increasing levels of noise, in addition to outperforming the benchmarked methods, which included 5 variations of the SMOTE technique, particularly as noise levels continued to increase.

The literature suggests that the class imbalance challenge has received more research attention than the other identified HAR challenges, as a larger amount of previous works have been found with the aim of addressing the class imbalance problem. According to [153], the number of research efforts pertaining to class imbalance has been continually increasing.

2.6.4 Recognition of Interleaved and / or Concurrent Activities

The recognition of concurrent and interleaved activities are complex challenges associated within HAR. Many approaches to HAR make the assumption that individuals perform activities sequentially, one at a time, however, approaches to HAR in a more naturalistic, real-world setting are anticipated due to the act of multitasking existing as an inherent characteristic in unsupervised daily routines [157]. Daily activities performed by inhabitants in naturalistic settings are very often concurring, interchanging, and occasionally abandoned. Concurrent activities describe those in which several activities are performed simultaneously, whereas interleaved activities describe those performed in an interwoven manner [157]. Figure 2.9 presents examples of both interleaved and concurrent activities, in which interleaved activities involved cooking, then pausing the cooking activity to answer the telephone, and subsequently resuming the cooking activity. The example of concurrently performed activities presented involved reading a book whilst drinking coffee at the same time.

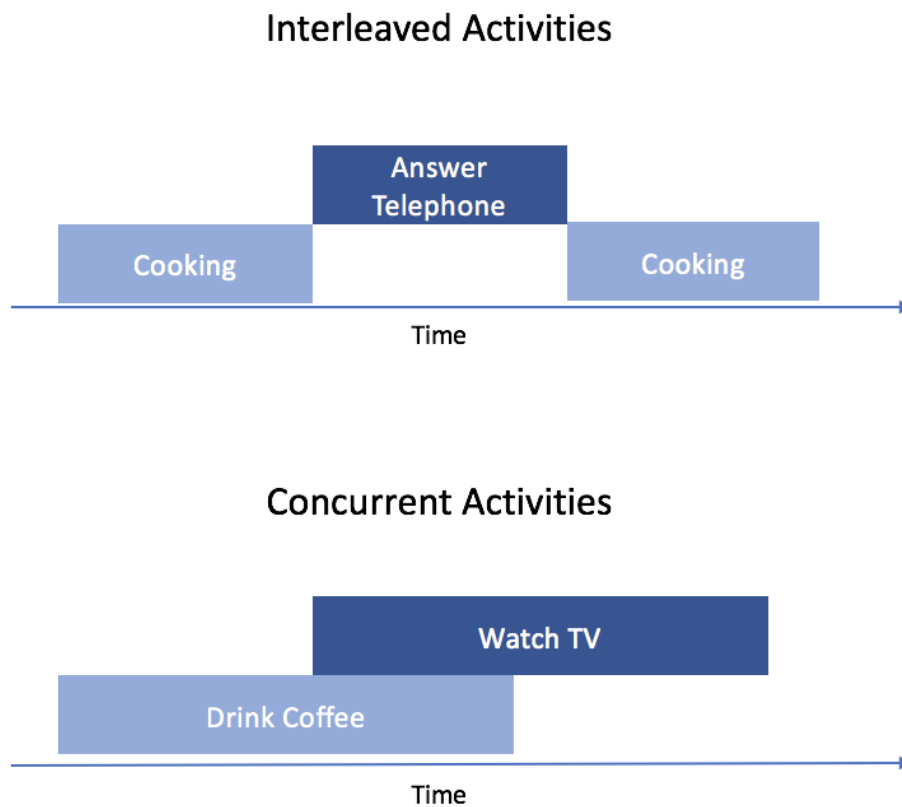


Figure 2.9. Examples of interleaved and concurrently performed activities, adapted from [157]

Previous efforts have been made in handling interleaved and concurrent activities. In a study conducted by [157] this problem was addressed with the proposed data-driven approach. A Strong Jumping Emerging Patterns (SJEP)-based feature discovery method was implemented to distinguish both simplistic and intricate activities in Phase 1 of the framework. Subsequently, fuzzy set theory was introduced in Phase 2. The proposed approach was evaluated upon 2 datasets, namely the CASAS Interleaved Activities of Daily Living and the SICA (Sequential, Interleaved, and Concurrent Activities) dataset. The SJEP method was compared against well-established classifiers, namely SVM, kNN, NB, HMM, and DT models. Experimental results had proven the effectiveness of the implemented method on both datasets through outperforming all other considered methods, whilst also demonstrating its supremacy in comparison to previous state-of-the-art approaches. Further to this, a study conducted by [158] addressed the problem of handling concurrent and interleaved activities through activity profiling within the data pre-

processing phase. For example, expert knowledge pertaining to specific activities and their associated sensors were defined to enable the recognition of activity sub-tasks occurring out of sequence. Temporal relationships were also considered to provide additional activity context. Following this, a dynamic windowing method was implemented to adaptively select appropriate window sizes per activity. The proposed approach was evaluated on 3 HAR datasets using 3 well-established classifiers, namely NB, SVM and DT models. Experimental results of this study demonstrated substantial performance improvements in recognising interleaved and concurrent activities through implementing the proposed approach, in comparison to the baseline method applied.

2.7 Conclusion

This Chapter has provided a review of the literature regarding HAR. As stated, due to the identified limitations of knowledge-driven approaches, the work in this Thesis will focus on data-driven approaches to HAR. Particularly, sensor-based HAR will be the focus of this work to eliminate privacy concerns and ethical issues, whilst also being lightweight and relatively low energy. This Chapter has identified HAR application domains, which predominantly included Smart Environments. A number of dedicated Smart Environments were identified which focus on producing and exploring AAL scenarios, for example the CASAS project, which also provide publicly available HAR datasets for the research community to avail of to conduct their own investigations.

Through reviewing the literature, it appears the more prevalent method of acquiring data is through body-worn sensors, rather than through environmentally deployed sensors. According to [17], the ease of access to wearable sensors and their relative low cost may have made them more appealing than more expensive environmental sensors. Nevertheless, publicly available, environmental sensor-based HAR datasets exist, and will be explored. Through explorations of the literature in this Chapter, it was also found that data pre-processing is an imperative aspect to consider during the HAR process, as data-driven approaches are reliant upon good quality data to achieve optimal classification. Furthermore, according to [18], large scope for investigation into data quality exists, including explorations into

noise detection and handling mechanisms. Another important aspect to consider within the HAR process is feature selection, according to the reviewed literature. Previous efforts seem to have focused more on features derived through wearable sensor signals, whereas it has been acknowledged that relatively less effort exists on feature selection based upon environmental sensors, such as binary sensors. Additionally, hybrid feature selection techniques have emerged more recently, indicating an opportunity for further exploration. A range of well-established classifiers were discussed, as well as the recent emergence of ensemble methods. Ensemble methods have revealed promising results in comparison to individual classifiers, thus also demonstrating an opportunity for further exploration. Finally, a number of HAR challenges were identified and research efforts into these areas were reviewed, including the problem of intraclass variability, interclass similarity, class imbalance and the challenge of recognising interleaved and / or concurrent activities. It was recognised that class imbalance had received considerable research attention in comparison to the remaining identified challenges as more previous efforts pertaining to this had been discovered.

Chapter 3

The Impact of Dataset Quality on the Performance of Data-Driven Approaches to HAR

3.1 Overview

Within the realms of data-driven approaches to HAR, data quality is a significant consideration. Nevertheless, data quality considerations are seemingly disregarded quite regularly as many researchers deliberate more specifically on the classifiers and techniques, whilst assuming the quality of data is sufficient [159]. Consequently, it is a common occurrence that data-driven classifiers are constructed

using low quality, suboptimal data that adversely impacts upon their performance [160].

This Chapter discusses and presents the impact that data quality has on activity classification using data-driven approaches. These approaches rely on good quality training data which has been the motivating factor for this study. A range of data-driven classifiers have been applied to generate models for activity classification, where the importance of data quality is highlighted by analysing the effects of noisy data through comparing the classification performances of raw (noisy) and subsequently cleaned data. A secondary factor motivating this work involved addressing the importance of developing a clear HAR data collection protocol to ensure the prevalence of noise and outliers are minimised. Data collection is becoming a critical consideration amongst the numerous challenges in the HAR domain, and it is recognised that a large majority of effort and time consumption spent in this domain is focused on data preparation, which involves acquiring, cleaning and interpreting the available data [80]. Furthermore, according to a recent review [18], explorations into noisy data have significant scope for further research.

The remainder of this Chapter is presented as follows: Section 3.2 generally describes data quality, Section 3.3 describes data cleansing within machine learning, including types of noise that may emerge within the data, Section 3.4 details the methodology undertaken to investigate the impact of noise within HAR data, and Section 3.5 presents experimental results and discussion. Following this, Section 3.6 concludes this Chapter.

3.2 Data Quality

Assessing the quality of data for machine learning tasks, particularly data-driven approaches, is a vital consideration. Data quality may be evaluated using various measures, including accuracy, completeness, uniqueness, consistency, timeliness and validity [161]. Evaluating the accuracy of data involves assessing whether it is correct or incorrect, for example determining whether data values reflect their related objects. Within the realms of machine learning for HAR, accuracy

issues may arise, for example, with incorrect annotation of the activity data in which an incorrect class label may have been assigned to a specific data instance during acquisition, thus adversely affecting performance attained during classification. The completeness factor relates to whether all expected data has been recorded and is therefore entirely present, whereas the consistency measure involves evaluating whether data is consistent amongst all data stores and systems. For example, inconsistencies in naming conventions may occur during HAR data collection when several participants are involved in the data acquisition process, such as class names being recorded as “1,2,3” by one participant, and “A,B,C” by another. The uniqueness measure ascertains whether data records can be uniquely identified through ensuring no duplicates exist, whereas the data validity measure determines whether data abides by any procedures and policies in place, for example, assessing whether data collected is truly representative of the phenomenon measured through determining a range of acceptable values and ensuring the data analysed has conformed with this acceptable range. Data timeliness involves evaluating whether data is up to date and available at the time required, for example determining whether the necessary data is accessible at the time expected to be processed.

3.3 Data Cleansing

Data cleansing can be defined as the process of removing errors or inconsistencies such as noise and/or outliers from a dataset [162]. The presence of noise and outliers in data is an important issue to address as these can have a substantial influence on the results produced by data-driven techniques, according to [163]. Nevertheless, [164] states the border between normal and abnormal (noise/outlier) data is often unclear, where a large “*grey area*” may exist.

3.3.1 Outliers

The presence of outliers in data structures represent observations that exist far from other data values such as an abnormally large distance from other observations. For example, outliers may emerge as arbitrary values that do not

conform with the underlying trend. Within most domains, expected data values possess a “normal” model, where deviations from this model are identified as abnormal, thus outlier detection methodologies output either an outlier score, or a binary label indicating whether data samples are normal or abnormal [165]. The outlier score determines the level of abnormality of each data sample evaluated, which can then be ranked. Ascertaining the level of abnormality reached to consider the data as an outlier is often a subjective decision [165]. Additionally, weak or strong outliers can be determined. Weak outliers often identify as noise such as data samples existing outside the defined ad-hoc threshold of normal data, whereas strong outliers typically possess a much larger score [165].

Many clustering tools exist to detect outliers, such as the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) tool. DBSCAN is valuable in handling clusters and outliers in that clusters may be established as arbitrary shapes, sensitivity to noise and outliers is substantial, and simplistic parameters are involved [166]. Other outlier detection methodologies include the use of probabilistic and statistical models [13], [167], linear regression models [165], or other distance-based models [168] where the identified outliers are subsequently removed. For example, in [167] researchers proposed calculating the absolute deviation around the median (MAD) as an alternative to the commonly implemented method of calculating the standard deviation around the mean, due to identified limitations of the latter approach such as the assumption of normal distribution, and the problem that outliers are very unlikely to be detected within small samples of data whilst utilising standard deviation around the mean. Further to this in [165], linear models were explained, where linear correlations are utilised to allow for data modelling along low-dimensional subspaces. An optimal hyperplane is determined to represent normal data; thus, outlier scores can be determined for data samples that deviate far from this normal data. In [168] common distance-based outlier detection methods were explained, which included defining outliers as those data samples whose distance pertaining to a k -nearest neighbour was largest, or whose average distance in relation to a k -nearest neighbour was largest, for example. In [169], outliers were detected through the Boxplot method, which discovers outliers and portrays data distributions based upon the following criteria: minimum value, maximum value, lower quartile, median value and upper quartile. Further to this, in a recent HAR study conducted by [170], an outlier detection method based on deep

recurrent NNs was proposed to detect outliers within sensor data derived through wearable devices. Outliers were detected within two HAR datasets comprising activities such as walking, running, ascending stairs and descending stairs. The proposed method involved 2 stages of outlier detection. Stage 1 involved data cleansing during pre-processing to remove initially detected outliers and stage 2 involved training the deep recurrent NN and performing final, enhanced outlier detection. Promising results were attained proving the effectiveness of the proposed method in detecting outliers, which outperformed the benchmark comparative studies by 3%.

3.3.2 Missing Values

As stated, missing values refer to the completeness measure in data quality. Incomplete data pertains to attribute noise as the expected data does not exist, for example a missing attribute value within a dataset due to equipment failure, or human negligence in entering data due to uncertainty or misinterpretation, thus omitting the required value [161]. Various methods of handling missing values exist, including either removing the entire record of data, or imputing the missing values based upon other data observations. Removing an entire record of data is not recommended, as there may be a large amount and therefore the complete dataset would be drastically reduced in size. Additionally, the remaining data within the record may be of good quality, thus if the entire record was removed, a large quantity of valuable data would be unnecessarily omitted [19]. Consequently, imputing the missing values is preferential. Handling missing numeric data may involve imputing a value based upon the attribute mean or median, whereas handling missing categorical data may involve adding a new category, such as a global constant as a new class [161]. For example, a common occurrence in large, free-living smart environment datasets is to impute a new “other” category to describe instances without an assigned class label as this may be data occurring between target activities. Many useful, open source tools exist in dealing with missing values, for example the Python Data Analysis (PANDAS) library is a powerful data analysis and manipulation resource utilised to improve data quality. In a study conducted by [43], a well-established HAR dataset, namely Aruba, generated within the CASAS smart home was processed for the

purpose of improving the following stages of the HAR process: feature extraction and classification. It was recognised that over half of the dataset contained unlabelled activity instances, thus the researchers labelled this data as “other” during the pre-processing stage. Furthermore, the Aruba dataset was processed within [85], within which the researchers also described the unlabelled activity instances as “other”.

3.3.3 Inconsistent or Incorrect values

The presence of inconsistent or incorrect values within data relate to the aforementioned consistency and accuracy data quality measures. For example, inconsistent data may comprise of discrepancies in naming conventions or representative codes, as developing an “A, B, C” convention within one data store could be unintentionally adapted to “1, 2, 3” in another, thus resulting in inconsistencies across data stores [161]. Incorrect values involve any data that does not conform to their related objects, for example, if a Surname is entered into a Title field within a data record. Data recognised as inconsistent or incorrect may be enhanced through the utilisation of data quality tools. A number of open source tools exist for improving data quality, such as OpenRefine. This visualisation and manipulation tool supports both the detection and correction of data inconsistencies through transforming data from one format to another. For example, incorrect text inputs can be identified through using a Text Facet within OpenRefine, which allows a user to view data more simplistically, such as data within a specific column, through displaying their unique inputs along with the frequency of their occurrence. Consequently, potentially incorrect inputs can be visually identified. Additionally, incorrect values may be amended or removed. In a study conducted by [171], a platform aimed towards supporting the independence of elderly smart home inhabitants was proposed. The HAR dataset utilised contained 16 ADLs gleaned through over 80 environmentally deployed reed switches. Data pre-processing was emphasised during this study, including the process of data transformation to ensure consistency within the data. For example, all date formats were transformed to achieve consistency.

3.4 Methodology

The dataset utilised in the current study was collected by 141 students enrolled in the Pervasive Computing in Healthcare module at Ulster University using a triaxial accelerometer [13]. The students were each assigned a HAR scenario containing 3 activities, and were subsequently tasked to collect, process, and classify data as part of the module assessment. In total, there were 6 scenarios and 18 activities recorded amongst the cohort. Table 3.1 presents details of the dataset, including the number of participants assigned to collect activity data per scenario.

Table 3.1. Dataset description including the number of participants assigned to each HAR scenario, and the activities involved within each of the 6 scenarios

Scenario	No. of participants	Activities
Self-Care	24	Hair grooming, brushing teeth, washing hands
Cardio	23	Walking, running, jogging
House Cleaning	25	Wash windows, ironing, wash dishes
Food Preparation	23	Mixing food, chopping veg, sieving flour
Sports	25	Pass, catch, bounce
Weights	21	Arm curls, lateral arm raises, deadlift

To investigate the impact of noise on HAR performance, a subset of the data was considered which consisted of the Self-care scenario involving hair grooming, hand washing and teeth brushing activities. This scenario was chosen for initial consideration as [13] states hair grooming and hand washing activities were most difficult to discriminate between of the 18 activities recorded. The selected scenario contained recordings produced by 24 participants, thus there were 72 activity files as each participant individually recorded the 3 activities specified in the self-care scenario.

3.4.1 Data Acquisition

Data utilised for experimentation in this Chapter was previously collected as reported in [13]. As previously mentioned, data was collected using a triaxial accelerometer, specifically a Shimmer device placed on the participants' dominant wrist to record data for each activity, presented in Figure 3.1. The students were instructed to follow a data collection protocol and were given video examples clearly demonstrating how the activities should be recorded, presented in Appendix 1. Students were also provided with well-defined guidelines as to how the sensor should be configured and calibrated, as data acquisition was performed unsupervised.



(a) Shimmer device with the X, Y, and Z axis displayed



(b) Shimmer device attached to wrist

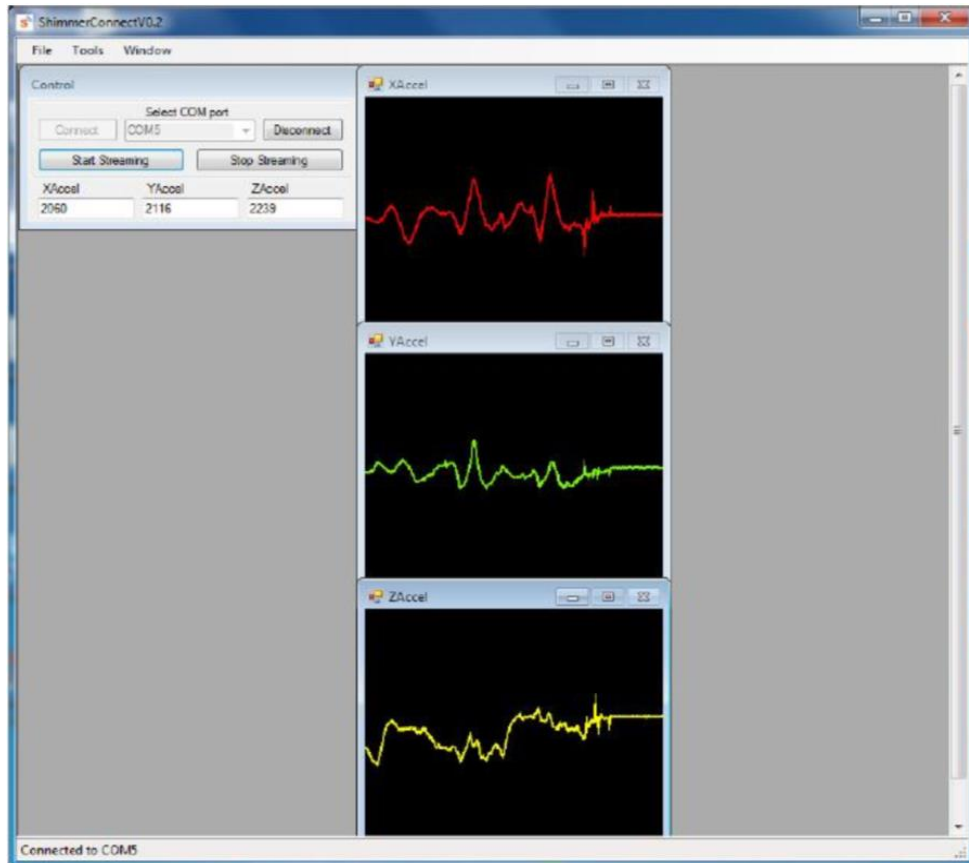
Figure 3.1. Shimmer device where (a) presents the Shimmer device axis, and (b) presents an example of the correct Shimmer device placement on a wrist.

The Shimmer device was initially to be calibrated using the Shimmer 9DOF Calibration software, presented in Figure 3.2, to determine the sensitivity and offset of the sensor. The offset depicts the sensor value observed when the true value is zero, whereas the sensitivity is determined by the chosen accelerometer range. The smaller the range, the more sensitive the sensor will be. Considering the triaxial accelerometer utilised, 6 observations were required to calibrate the X, Y and Z axis, as each axis possesses two degrees of freedom.

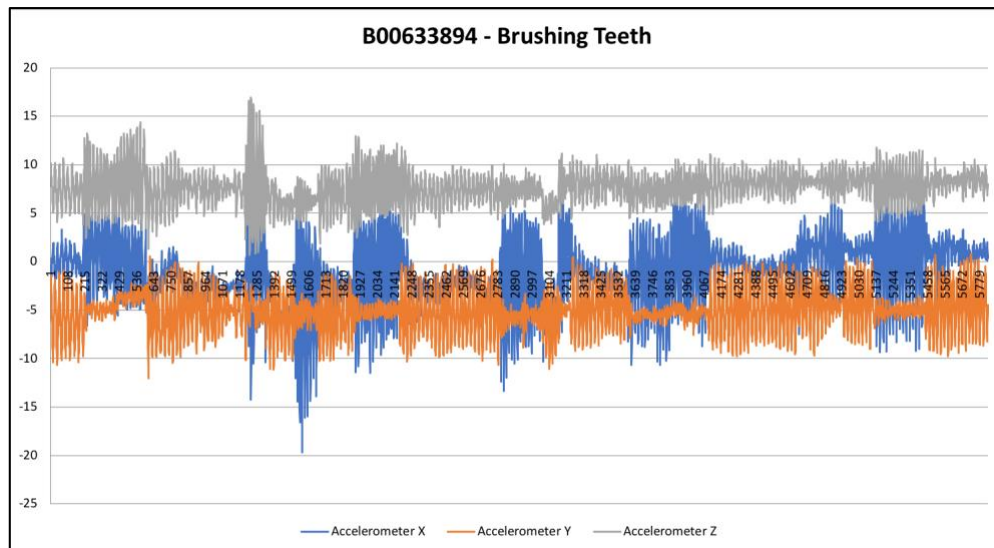


Figure 3.2. Shimmer Calibration software

Data was collected using the Shimmer wireless sensor platform. Prior to recording, the device was configured with a sampling rate of 51.2Hz and a sensitivity range of $\pm 1.5g$. During each activity recording the data for the X, Y and Z axis is streamed and stored in a .csv file format along with a timestamp. Due to this, approximately 6120 samples were expected to be recorded per activity, per person as each of the participants were tasked to collect 2 minutes of data per activity. Figure 3.3 presents the Shimmer Connect software used to record activity data, demonstrating movement along the X, Y and Z axis, along with an example of the full activity recording produced through the teeth brushing activity.



(a) The Shimmer Connect software used to record activity data



(b) A full activity recording example of the teeth brushing activity performed by one participant

Figure 3.3. (a) presents an example of the Shimmer Connect software used to record activity data from the accelerometer and (b) presents a full example of the produced activity recording (approximately 2 minutes duration).

3.4.2 Data Cleaning

As presented within Section 3.3, many established tools and techniques exist for the purpose of analysing and enhancing the quality of data, as well as recognised methods in detecting noise. It was decided to conform with a manual cleaning technique for data quality explorations within this work, recommended within the Pervasive Computing in Healthcare module offered at Ulster University, presented in Appendix 2. This technique involved visualisation of the accelerometer signals for manual inspection and quality enhancement. The purpose of visually inspecting activity data was to provide support in gaining valuable experience, whilst also developing knowledge, upon accelerometry-based data validation.

Thus, activity data was visually inspected through plotting the individual graphs of the participants recorded data files to ensure that the data collection protocol had been adhered to and also to identify potential noisy portions of recordings. The data was cleaned based on errors (noise) introduced by participants that recorded data incorrectly, which can be easily visually identified. The purpose of data cleaning was to detect outliers, for example, random large spikes and issues such as poor calibration of the Shimmer device, or a sensitivity range outside of the measurable capacity of the sensor. Additionally, each participant's activity files were examined to check for the possible existence of brief time delays between starting/ending the recordings, and performing the target activity as these portions of data are not representative of the target activity. Figure 3.4 demonstrates the steps taken to assess the quality of each data recording, within which the sources of noise are either handled by discarding the activity file or enhancing its quality by removing noisy portions of data, thus resulting in a sufficiently representative activity recording to proceed to the next stage of the HAR process.

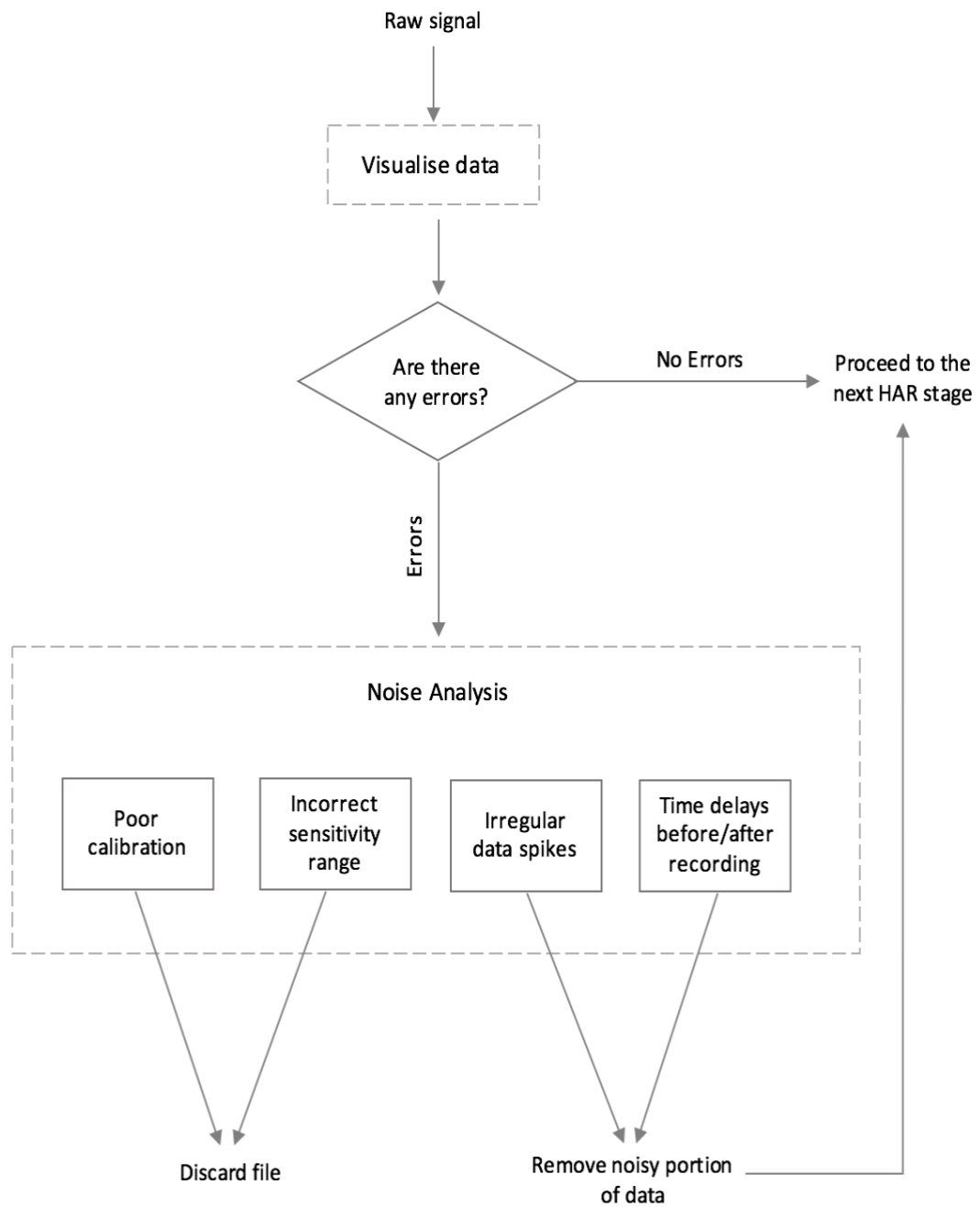


Figure 3.4. Data cleaning process within which sources of error (noise) are identified and handled

Figure 3.5 presents an example of removing time delays from an activity recording, indicating the removal of noise as this additional data is irrelevant to the activity being performed and may cause confusion during classification. This is an example of ensuring data validity, as the time delayed portions of data are not representative of the activity being recorded, thus, this data is invalid.

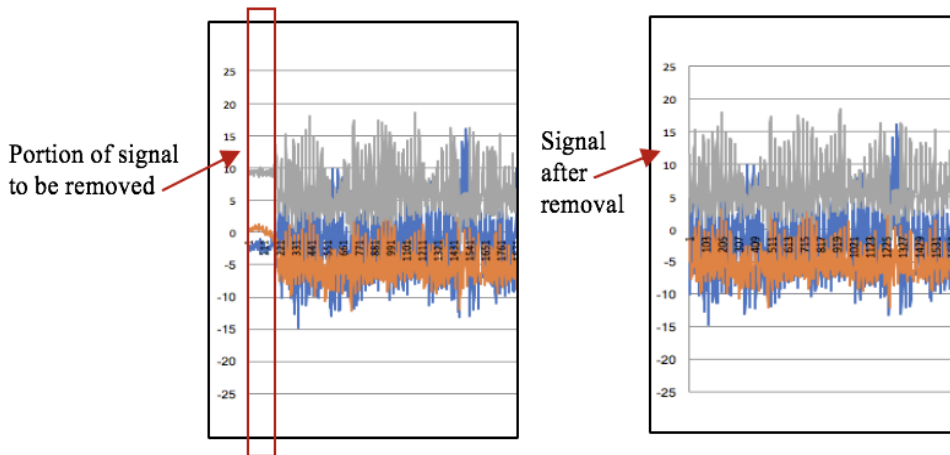


Figure 3.5. Displaying a signal before and after the removal of noise, caused by a time delay at the beginning of a recording.

As mentioned, outliers which were clearly distinguished during data inspection were removed to attain a better quality signal. Figure 3.6 presents an example of a random spike existing in an activity file, which was subsequently removed from the file. This is an example of an outlier, as the data spike presented in Figure 3.6 has largely deviated from the data trend observed, demonstrating this abnormal data value exists far from all other values recorded.

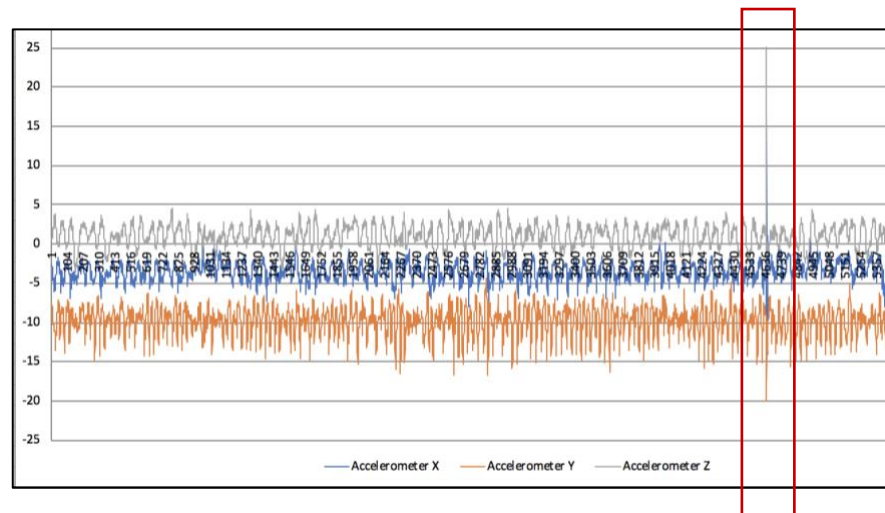


Figure 3.6. Random large spike to be removed from the signal

A large majority of data issues detected, specifically noisy portions of signals, could simply be removed from the activity data to improve quality, as the remainder of these signals possessed good quality, representative activity data. Certain activity files were fully discarded from the dataset due to issues such as a signal having a range outside the measurable capability of the sensor. For example, if the range was correctly set during configuration to $\pm 6G$, and the target activities were correctly recorded, an abnormally large axis would not exist on the activity graph, as presented in Figure 3.7. This is an example of discovering invalid data, as the data being recorded is not representative of the phenomenon being measured (the target activity).

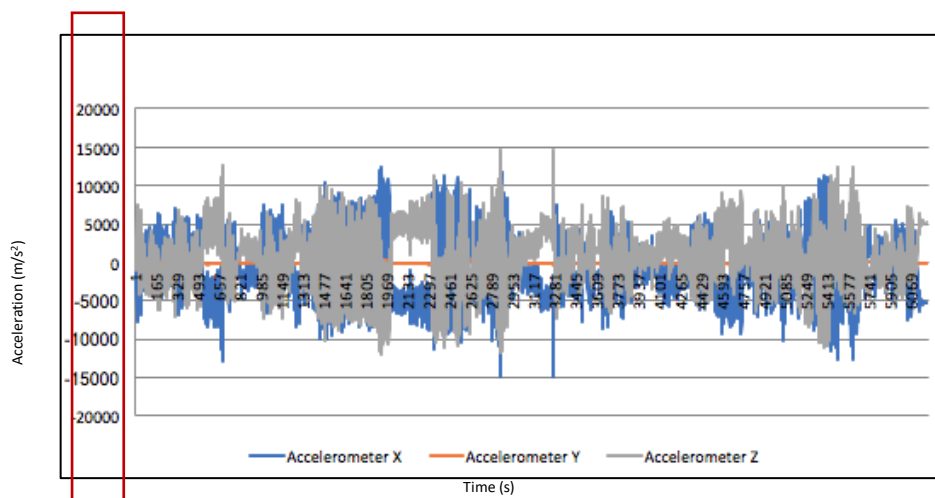


Figure 3.7. Displaying a signal with a range outside the measurable capability

As a result of cleaning the data, 9.44% of instances were removed from the dataset as these were deemed noisy or erroneous samples (the raw dataset contained 2891 instances before cleaning, and subsequently contained 2618 instances following this).

3.4.3 Segmentation

Data was segmented using time-based windowing to identify segments of the data streams that were likely to contain information regarding activities. Time windowing is a commonly utilised technique pertaining to accelerometers where sensor data is sampled at a sustained rate [9], although no clear consensus has been identified as to which window size should be implemented [172]. If the window size employed is too small, there may not be enough relevant activity information obtained to make a beneficial decision, however, if the window size is too broad, the measurements made may have too few variations to result in appropriate decisions for activity classification [173]. A non-overlapping window size of 3 seconds was deemed appropriate as [172] states energetic activities such as walking, jogging and running can be optimally detected between 1 and 3.25 seconds, while more complex activities may require longer time windows.

3.4.4 Feature Extraction

A compilation of standard time domain features were extracted from the windowed data to obtain relevant information and to represent the characteristics of the various activity signals. Extracted features included the mean, maximum, minimum, standard deviation, variance, root mean square (RMS), signal magnitude area (SMA), range, and median for the X, Y, and Z axis and signal magnitude vector (SMV), and the cross correlation for each axis, as [70] suggests these features are suitable for activity recognition. A total of 39 features were extracted, as presented in Table 3.2.

Table 3.2. Features extracted from the windowed data

Feature No.	Feature Name	Description
1-4	Mean	Mean value of the X, Y, Z and SMV
5-8	Maximum	Maximum value of the X, Y, Z and SMV
9-12	Minimum	Minimum value of the X, Y, Z and SMV
13-16	Standard Deviation	Standard deviation of the samples X, Y, Z and SMV
17-20	Variance	Variance of the X, Y, Z and SMV
21-24	Root Mean Square	Root mean square of the X, Y, Z and SMV
25-28	Signal Magnitude Area (SMA)	SMA across the acceleration signal in X, Y, and Z axis
29-32	Range	Range of the samples of SMV in the window
33-36	Median	Median of the X, Y, Z and SMV
37-39	Cross Correlation	Cross correlation of the X, Y and Z axis

The statistical features mentioned are common due to their simplistic nature and significant performance across a variety of activity recognition problems [70]. The maximum, minimum and range features can assist in differentiating between activities that contain movements comprised of different ranges [74]. SMA has proven beneficial when employing triaxial accelerometers for activity recognition as it can suitably differentiate between static and dynamic activities [174]. SMV signals are independent of the orientation of the sensor [175] and were consequently valuable to include as the dataset contained a large number of participants, each placing the sensor on their dominant wrist.

3.4.5 Activity Classification

Four standard classifiers were chosen to make decisions on the activities being performed, namely NB, DT, kNN and an MLP. The classifiers were constructed using Weka and configured using the recommended default parameters within Weka. These classifiers were chosen as NB, DT and kNN are recognised within [176] as effective classifiers, and MLP is recognised as effective for HAR tasks according to [47], [177]. 10-fold cross validation was used for training and testing the models, and classification accuracy was measured to assess model performance. Classification accuracy is calculated as the number of correctly classified instances divided by the total number of instances. Model performance was first evaluated on the raw data with the four classification algorithms, where both the training and test sets included noise (N-N). Following this, the four classification models evaluated were retrained with cleaned data through applying the cleaning method described in Section 3.3.2. In this case, both the training and test sets consisted of data with improved quality (C-C). Nevertheless, to simulate a real-life application another case was introduced to ascertain whether the models were able to retain their capability of generalising; consequently, each model was trained on a cleaned set and tested on a noisy set (C-N). Another case was initially considered where the models would be trained on noisy data and tested on cleaned data (N-C), however this was not employed as the N-C combination was deemed to be an invalid scenario for evaluation. Significance testing was applied to determine whether the comparisons made were of statistical importance using t-testing with a 95% confidence. Thus, a p -value of less than 0.05 was believed to be statistically significant.

3.5 Results and Discussion

The obtained results demonstrate the importance of conducting a data quality assessment during the data preparation stage in relation to HAR, particularly when the acquired activity data has been collected via a large number of unsupervised participants. Whilst implementing the data cleaning methodology described in

Section 3.4.2, various sources of noise were identified and subsequently rectified, thus, the obtained results reflect the adverse impact these issues had on classification performance when comparing the original data to the improved quality data. Sources of error that were identified and removed, which evidently had an impact on classification performance, predominantly included the removal of time delays, for example, portions of recordings with non-representative data incorrectly recorded before or after performing the target activity, random spikes existing within activity files, and issues such as having a signal range outside the measurable capability of the employed accelerometer. Table 3.3 provides the classification accuracies produced by the four evaluated algorithms for the activity recognition problem, within which N-N indicates a noisy training set paired with a noisy test set, C-C indicates a cleaned training set paired with a cleaned test set, and C-N indicates a cleaned training set paired with a noisy test set. The kNN model achieved consistently superior performance across all considered cases (N-N, C-C, and C-N), whereas the Naïve Bayes model performed least effectively, in comparison to the remaining evaluated classification models.

Table 3.3. Accuracies of four algorithms for the classification of activities included in the Self-Care scenario.

	N-N (%)	C-C (%)	C-N (%)
MLP	89.688	90.939*	89.272
DT	84.516	85.935	85.502
kNN	90.862	92.202*	92.522**
NB	67.240	80.207***	75.887***

*Note: The table displays comparisons between whether there was a significant difference N-N & C-C cases, and N-N and C-N cases through T-Testing. * $p < 0.05$. ** $p < 0.01$. *** $p < 0.001$.*

The results in Table 3.3 demonstrate the effectiveness of the data cleaning methodology undertaken, as cleaning the data through visual inspection lead to an

improvement in performance of all four classifiers when generating comparisons between the N-N & C-C cases, with significant improvements made by the MLP, kNN and the NB models. The NB model improved most substantially in terms of classification accuracy, demonstrating an increase of 12.967% from 67.240% to 80.207%, whilst comparing the N-N to C-C cases. The remaining three models, namely kNN, DT, and MLP improved by 1.340%, 1.420%, 1.250%, respectively. This outcome proves the benefits of training and testing models on cleaned data and suggests that noise has an apparent negative impact on classification accuracy.

In terms of robustness to noise, the results produced by each classifier were individually analysed. For example, considering the NB model had improved most substantially between the N-N and C-C cases, which indicated its poor ability to handle the impact of noise. This may be due to the classical NB model being sensitive to outliers, as reported in [104]. Some studies suggest the NB model is relatively robust to noise due to its conditional independence assumption (naivety), in relation to the various input features [178], [179], yet it was also stated that model robustness may be dependent upon several factors such as the level of noise present within the dataset, as well as the characteristics of the data, for example, whether class imbalance exists. Nevertheless, the obtained results in Table 3.3, along with investigations conducted in [104], indicate poorer robustness to attribute noise. Furthermore, the DT and MLP models performed moderately well in terms of robustness to noise during the experiments conducted. According to [110], [180], an advantage of DTs include their tolerance of outliers. Nevertheless, it was also mentioned in [110] that overfitting may occur, particularly whilst evaluating a small dataset, which may adversely affect its ability to handle noise. The overfitting problem is also commonly acknowledged whilst implementing NNs, as they are able to learn complex decision boundaries during training. Yet, the obtained results have demonstrated an avoidance of overfitting, as both the MLP and DT models had retained their ability to generalise. Finally, the kNN model initially demonstrated sufficient robustness to noise through obtaining the optimal classification performance before implementing the data cleaning methodology. This conforms with [102], [181], which stated an advantage of the kNN classifier is its ability to tolerate noisy data. Additionally, whilst comparing the N-N and C-C cases, the kNN classifier achieved a small increase in performance which indicates that whilst

demonstrating adequate robustness to noise, the adverse effects of noisy data remained.

A more realistic case to consider is the C-N circumstance, as a model can be trained on cleaned data, though the prevalence of a clean test set is highly unlikely in a more naturalistic, real-world setting. This case evaluated whether the models were capable of generalising if tested with noisy data. When generating comparisons between the N-N & C-N situations, the NB and kNN classifiers improved significantly and retained their ability to generalise. Again, the NB model demonstrated the largest improvement in classification accuracy, increasing from 67.240% to 75.887%. The MLP and DT did not significantly improve, yet they did not perform worse or lose their generalisation capability, even after being trained on a smaller dataset (as 9.44% of instances were removed during cleaning). As previously stated, there may be an unclear border between normal and abnormal data, which indicates some instances may have been removed that were seemingly outliers though may have actually been useful to remain within the dataset; therefore, the MLP and DT may have been affected as a result.

The obtained results indicate that a revised and refined data collection protocol would benefit the quality of the diverse dataset gleaned by a large number of unsupervised participants, as the identified data issues adversely impacting HAR classification performance were introduced during data collection and may have been avoided. For example, consider the brief time delays that had occurred before/after performing the target activity during some recordings that were irrelevant and unrepresentative of the activity. Through examining the data collection protocol, it had been observed that during the calibration stage, participants were instructed to leave the sensor untouched for 5 seconds prior to calibrating each orientation. This may have caused some confusion to participants during the activity recording stage, in which they may have assumed a 5 second time delay was also required at this stage. Thus, the data collection methodology should be updated to clearly distinguish no time delays should occur during activity recordings. Additionally, calibration and configuration of the accelerometer was clearly defined within the protocol yet issues still transpired. Therefore, the protocol should be updated to emphasise the commonly occurring data collection issues to ensure participants are particularly attentive in avoiding those. For example, ensuring the correct sensitivity range is chosen during configuration.

3.6 Conclusion

As stated, data quality is an imperative consideration pertaining to the classification performance of data-driven classifiers. Thus, data gleaning is also becoming a critical concern in the realms of machine learning as noise and outliers are commonly introduced during this stage. Noise and outliers may particularly occur in unsupervised, large-scale data collection scenarios, in which participants may introduce avoidable data issues due to lack of understanding or naive negligence, thus leading to suboptimal activity recordings.

The impact of data quality on activity classification using data-driven models has been evaluated during experimentation in this Chapter. An effort in generating performance comparisons between noisy and cleaned data for HAR has been presented, using a diverse dataset collected by multiple participants in an unsupervised setting. Data was gleaned using a triaxial accelerometer located upon the dominant wrist of each participant and segmented through time-based windowing. Additionally, 39 features were extracted to represent the characteristics of activity signals, subsequently evaluated with four common data-driven classifiers, namely DT, NB, kNN and MLP. As ascertained, data-driven approaches rely on good quality data which has been demonstrated through the experimental results obtained. The data cleaning methodology undertaken had demonstrated its effectiveness as various sources of noise were discovered and removed, which enhanced the classification performance achieved by all four classifiers. The performance of each classifier was evaluated on the raw data, specifically the N-N case, and subsequently evaluated on the improved quality data, namely the C-C case. Lastly, the C-N case was evaluated to simulate a real-life scenario and to ascertain whether the models were able to retain their capability of generalising. The NB model demonstrated the most significant improvements whilst generating comparisons in both N-N to C-C and N-N to C-N cases, followed by the kNN model.

Since noise was introduced during the data acquisition stage, experimental results highlighted the importance of following a data collection protocol attentively and recommend ensuring activity recordings contained high quality data for classification purposes. Additionally, findings from experimentation indicated that a further refined data collection protocol may be beneficial as the presence of noise

was largely due to participants failing to adhere to the data collection protocol attentively¹. According to [182], numerous challenges exist in collecting HAR data, which have currently led to low availability of publicly disseminated HAR datasets. Furthermore, the quality of these publicly available datasets is often unclear [81]. Some repositories, for example UCI Machine Learning Repository, contain some information concerning attribute types and missing values, nevertheless, data requires screening to confirm its suitability for classification [182]. Thus, Chapter 4 will assess the quality of a publicly available HAR dataset and prepare this data for use in succeeding Chapters.

¹ The results in this Chapter were published in [205]

Chapter 4

Recommendations for Pre-Processing Publicly Available HAR Datasets

4.1 Overview

As previously introduced in Chapter 3, data collection is recognised as a crucial concern amongst the numerous challenges in machine learning, largely due to limited amounts of training data being available to researchers in their respective fields, and the quality of the data being collected [80]. In the realms of machine learning, it is known that the majority of time and effort is consumed through data preparation, which includes gleaning, cleansing, and interpreting the data, as well as performing feature engineering [80]. The performance of data driven methods are

largely reliant upon the quality of data introduced during the training phase, thus, data quality is a vital consideration whilst implementing data-driven methods to HAR. Nevertheless, according to [13], the progression of HAR research is still being hindered by the scarcity of publicly available datasets that include a large quantity of accurately annotated and high quality data. Furthermore, according to [81], the need for investigations into developing high quality, accurate, refined pervasive health study protocols remains crucial in efforts towards providing more availability of shared research testbeds. Determining the reliability of collected data has been identified as a challenge regarding shared pervasive health datasets, with concerns raised pertaining to data collection and management [81]. Thus, the development of standards is required to ensure the effective sharing and re-use of pervasive health-related data [81].

This Chapter assesses the quality of a publicly available HAR dataset and prepares the data for use in succeeding Chapters. Section 4.2 describes the publicly available HAR dataset analysed within this Chapter, Section 4.3 details the preprocessing stage conducted, Section 4.4 provides recommendations for pre-processing data to ensure adequate quality, and finally Section 4.5 concludes this Chapter.

4.2 Dataset for Data-Driven HAR

Within the realms of data-driven HAR, a number of common data quality issues have emerged that adversely affect classification performance. These include annotation scarcity/incorrectness, issues with combining multimodal data sources, data inconsistency, heterogeneity of sensor data, and large imbalance of data existing within many HAR datasets [183],[13]. Due to the nature of human activities, class imbalance is a widely reported issue as some activities occur frequently throughout a typical day, whereas other activities occur occasionally. In a study conducted by [184] the adverse effects of imbalanced data were revealed in which classification performance was evaluated before, and subsequently after, the application of a resampling technique. Initial experiments demonstrated the adverse effects of imbalanced data as the minority classes, specifically transitional activities, performed much worse than the majority classes, specifically standard static and dynamic

activities. Subsequently, following the implementation of a resampling technique, classification performance had significantly improved. Thus, concluding that unbalanced data has a negative impact on performance during classification.

An overview of the explored HAR data is presented in this Section with emphasis upon the quality of data. The UCAmI Cup challenge is also described as the dataset utilised in this Chapter was derived from this competition. Section 4.2.1 outlines details of the original dataset, Section 4.2.2 highlights the problems identified and Section 4.2.3 details the restructured dataset created as a result of the encountered problems and to demonstrate more realistic capabilities of binary datasets for HAR in smart environments. The data used in this Chapter was generated for the 1st UCAmI Cup challenge, within which participants were encouraged to apply their tools and techniques to explore a HAR dataset with the ambition of attaining the highest classification accuracy upon an unseen test set. The challenge coordinators comprehensively describe the HAR dataset provided to participants in [79].

4.2.1 UCAmI Cup Dataset

The HAR dataset was collected over 10 days by researchers in the UJAmI Smart Lab [79]. The UJAmI Smart Lab is divided into five regions: an entrance, a workplace, a living room, a bedroom with an integrated bathroom, and a kitchen, which combined measures approximately 25 square meters, as presented in Figure 4.1. The dataset was captured and manually annotated by a single male inhabitant completing morning, afternoon and evening routines, representing 246 occurrences of 24 activity classes, as presented in Table 4.1. The training set consisted of 7 days of labelled data, with the remaining 3 days of data being provided as an unlabelled test set.

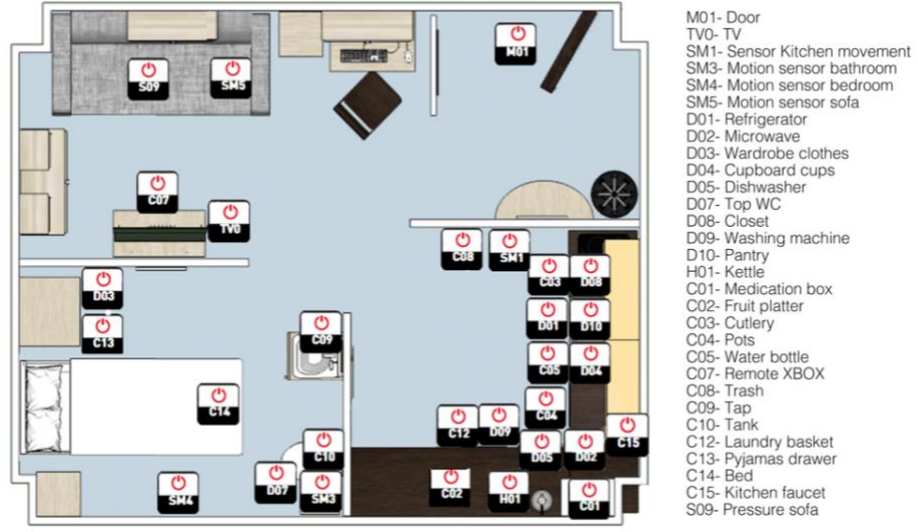


Figure 4.1. Location of binary sensors in the UJAmI Smart Lab [79]

Table 4.1. Activity classes in the UCAmI Cup dataset [60], where M, A and E indicate the Morning, Afternoon and Evening routines, respectively.

ID	Name	Instances	Routine
Act01	Take medication	52	A, E
Act02	Prepare breakfast	63	M
Act03	Prepare lunch	118	A
Act04	Prepare dinner	76	E
Act05	Breakfast	78	M
Act06	Lunch	101	A
Act07	Dinner	86	E
Act08	Eat a snack	12	A
Act09	Watch TV	70	A, E
Act10	Enter smart lab	21	A, E
Act11	Play a videogame	28	M, E
Act12	Relax on the sofa	85	M, A, E
Act13	Leave smart lab	33	M, A
Act14	Visitor to smart lab	7	M, A
Act15	Put waste in the bin	75	A, E
Act16	Wash hands	22	M
Act17	Brush teeth	132	M, A, E
Act18	Use the toilet	44	M, A, E
Act19	Wash dishes	13	A, E
Act20	Put washing in machine	20	M, A
Act21	Work at the table	20	M
Act22	Dressing	86	M, A, E
Act23	Go to bed	30	E
Act24	Wake up	32	M

A set of 30 binary sensors consisting of magnetic contact switches, PIR motion detectors, and pressure sensors were deployed in the UJAmI Smart Lab to capture human interactions within the environment, as presented in Figure 4.1. The two changeable states of the magnetic contact switches were open/close, which were attached to, or integrated within, doors and objects such as the medication box. The motion detectors generated and recorded movement/no movement states to identify whether an inhabitant had moved in or out of the 7-meter sensing range. Finally, the pressure sensors deployed generate either pressure/no pressure states and was present in the bed and the sofa to detect any interactions. A comprehensive description of each binary sensor is presented in Table 4.2.

Table 4.2. Description of binary sensors [35]

ID	Object	Type	States
SM1	Kitchen area	Motion	Movement/No movement
SM3	Bathroom area	Motion	Movement/No movement
SM4	Bedroom area	Motion	Movement/No movement
SM5	Sofa area	Motion	Movement/No movement
M01	Door	Contact	Open/Close
TV0	TV	Contact	Open/Close
D01	Refrigerator	Contact	Open/Close
D02	Microwave	Contact	Open/Close
D03	Wardrobe	Contact	Open/Close
D04	Cups cupboard	Contact	Open/Close
D05	Dishwasher	Contact	Open/Close
D07	WC	Contact	Open/Close
D08	Closet	Contact	Open/Close
D09	Washing machine	Contact	Open/Close
D10	Pantry	Contact	Open/Close
C01	Medication box	Contact	Open/Close
C02	Fruit platter	Contact	Open/Close
C03	Cutlery	Contact	Open/Close
C04	Pots	Contact	Open/Close
C05	Water bottle	Contact	Open/Close
C07	XBOX Remote	Contact	Present/Not present
C08	Trash	Contact	Open/Close
C09	Tap	Contact	Open/Close
C10	Tank	Contact	Open/Close
C12	Laundry basket	Contact	Present/Not present
C13	Pyjamas drawer	Contact	Open/Close
C14	Bed	Pressure	Pressure/No pressure
C15	Kitchen faucet	Contact	Open/Close
H01	Kettle	Contact	Open/Close
S09	Sofa	Pressure	Pressure/No pressure

4.2.2 Data Challenges

As stated, the data utilised within this Chapter was collected for the purposes of the 1st UCAmI Cup challenge [185]. It was reported that knowledge-driven, rule-based approaches outperformed the data-driven approaches to the activity recognition task, with several participants highlighting issues and limitations discovered within the data [39], [108], [186], [187]. The technique applied by [186] incorporated a domain knowledge-based solution inspired by a Finite State Machine, which achieved 81.3% accuracy on the unseen test set. In [108], a hybrid model was proposed using a hidden markov chain and logic model. The researchers combined their knowledge-driven and probabilistic models using a weighted averaging method, however, they reported they had expected a better result than 45.0% accuracy on the test set. Further to this, [39] used a Naïve Bayes approach with emphasis on location-aware, event-driven activity recognition. The applied method interpreted events as soon as they became available in real-time, omitting the need of an explicit segmentation phase, and generated activity estimations using an activity prediction model. Reported results show mean accuracies of around 68%, with the researchers stating that given the high number of activity classes, the outcome achieved was reasonable. Another approach implemented in [187] used various common machine learning algorithms, including a DT, kNN, SVM, and three ensemble approaches including a Random Forest, Boosting and Bagging. The researchers reported a training set accuracy of 92.1%, however, their approach achieved 60.1% on the provided test data which demonstrated poor generalization. Their suggested cause for the low outcome was the high imbalance of classes in the training set and stated the training algorithm required more labelled training data to perform better. Table 4.3 presents an overview of the techniques implemented along with the performances achieved by UCAmI Cup participants, and the reported data challenges.

Table 4.3. UCAMl Cup Challenge: Implemented techniques, performances achieved, and data challenges reported by participants

Publication	Implemented Technique	Train Accuracy (%)	Test Accuracy (%)	Reported Data Challenges
[39]	Recursive Naïve Bayes method with emphasis on location-aware, event-driven HAR	68	Undisclosed	Large number of activity classes
[108]	Hybrid knowledge-driven and probabilistic model using a weighted averaging method	Reported per routine: Morning 65.4 Afternoon 60.8 Evening 59.4	45.0	Imbalanced dataset and poor distribution of activities within train and test sets
[186]	Domain knowledge-based solution inspired by a Finite State Machine	Undisclosed	81.3	Low quantity of available data and missing sensor values
[187]	Comparisons of various classification models including DT, kNN, SVM, Random Forest, and ensemble methods	92.1	60.1	High imbalance of classes in the training set, and low quantity of available data.
[38]	Multi-Event Naïve Bayes Classifier using activity sequences and sensor events	68.0	60.5	Large number of activity classes
[188]	Random Forest classifier	94.0	47.0	Imbalanced dataset and poor distribution of the train and test sets

4.2.3 Acknowledged Limitations

The limitations and issues discovered within the original binary dataset that hindered classification performance whilst recognising ADLs in a smart environment setting, comprehensively included:

- Number of classes

The number of classes in the original dataset were very high given the low number of instances per activity and low amount of data overall. As discussed previously, data-driven approaches rely on large amounts of good quality data.

Furthermore, certain classes were too closely related to one another to recognise with binary data alone. For example, the following activities relied on one door sensor: Act 10 enter smart lab, Act13 leave smart lab, and Act14 visitor to smart lab. Binary sensors are limited in inferring activities in that they provide information at an abstract level [189], therefore Act08 eating a snack was difficult to distinguish compared to Act03 prepare breakfast, Act04 prepare lunch and Act05 prepare dinner as these activities all used similar sensors within the kitchen. Thus in order to capture activities at a finer level, the presentation and interpretation of binary data often requires further knowledge of the environment [190]. This issue was discussed by a UCAmI Cup participant in [39], where conclusions had stated that their achieved activity recognition performance was reasonable given the large number of activity classes present in the dataset.

- Imbalanced dataset

The distribution of instances per class in the original dataset were highly diverse, which may have caused minority classes to be overlooked by the classification model. For example, Act19 wash dishes was represented by 13 instances of data, whereas other activities such as Act17 brush teeth had more than 100 instances. Furthermore, the distribution of instances per class in the provided training and test sets were highly varied. For example, Act09 watch TV was very under-represented in the training set, yet the test set included a large number of Act09 instances. Noteworthy, Act09 watch TV also produced very similar sensor characteristics to Act12 relax on the sofa, which was problematic in the initial experiments as the training set included large amounts of Act12 data. This issue was discussed in [108] where researchers stated that their approach also found difficulty in classifying Act12 due to the poor representation of this activity in the training set, and suggested the data should be better distributed to improve HAR performance.

- Quantity of data

As previously stated, data-driven approaches require a large amount of data during the training phase to learn activity models, and to ensure these models can generalize well to new data. Thus, more labelled training data may have improved initial experiments. In [187], UCAmI Cup participants suggested the cause for their

low HAR performance was the high imbalance of classes in the training set, and stated the training algorithm required more labelled training data to perform better.

- Missing sensors

Act21, work at table, had no binary sensor located near the table to distinguish this activity, as demonstrated in Figure 4.1. This issue caused confusion as the sensor firing for Act21 in the labelled training set was seen to be a motion sensor located in the bedroom, which was irrelevant to Act21 and therefore seen as erroneous. In addition to missing sensors, there were also missing values from sensors that were expected to fire during certain activities. As previously stated, some researchers participating in the UCAmI Cup challenge reported they found missing values or mislabeling of some activities within the training set. In [186] this issue was discussed where participants stated that during one instance of Act10 enter smart lab, the only binary sensor that was expected to fire (M01), did not change states.

- Interclass similarity

This is a common HAR challenge that occurs when certain activities generate similar sensor characteristics, though they are physically different [70]. Table 4.4 presents the activities that produced similar sensor characteristics, resulting in difficulties arising in discriminating between these activities during classification.

Table 4.4. Activities producing similar sensor characteristics within the UCAmI Cup data

Activity Group	Activity Name	Common Sensors
Act10 Act13 Act14	Enter Smart Lab, Leave Smart Lab, and Visitor to Smart Lab	M01 Door
Act23 Act24	Go to Bed and Wake Up	C14 Bed
Act09 Act12	Watch TV and Relax on Sofa	S09 Pressure Sofa SM5 Sofa Motion
Act02 Act03 Act04 Act08	Prepare Breakfast, Prepare Lunch, Prepare Dinner, Prepare Snack	SM1 Kitchen Motion D10 Pantry C03 Cutlery

As a result of assessing the quality of data and others identifying various problems within the dataset, it was decided to restructure the data to reveal the potential of using binary sensors alone within smart environments.

4.2.4 Restructured Dataset of Improved Quality

The first step towards improving the quality of data by restructuring the dataset involved combining the provided training and test sets to better represent each activity class within the training data, thus striving to generate more robust models. Figure 4.2 presents the distribution of the combined 10 days of 24 activity classes for all the available data in the UCAmI Cup. As can be observed in Figure 4.2, some classes were exceedingly under-represented, with a third of all activity classes containing less than 30 instances. These classes were fully removed from the dataset as they would be highly under-represented during the training phase, and therefore would demonstrate poor generalisation to unseen data. Consequently, 8.82% of instances were removed, which comprised the following classes: Act08 eat a snack, Act11 play a videogame, Act16 wash hands, Act19 wash dishes, Act20 put washing in machine and Act21 work at the table.

An opportunity to combine certain similar activity classes emerged so that the data could be used effectively. For example, Act10 enter smart lab, Act13 leave smart lab and Act14 visitor to smart lab were combined to produce ActN1 door, as they all made use of a single door sensor, and Act09 watch TV and Act12 relax on the sofa were combined to produce ActN2 watch TV on sofa, as they mainly consisted of the inhabitant sitting on the sofa. Furthermore, Act02 prepare breakfast and Act05 breakfast, Act03 prepare lunch and Act06 lunch and finally Act04 prepare dinner and Act07 dinner were combined to produce ActN3 breakfast, ActN4 lunch and ActN5 dinner, respectively, as these sets of activities were similar. Table 4.5 presents the restructured dataset.

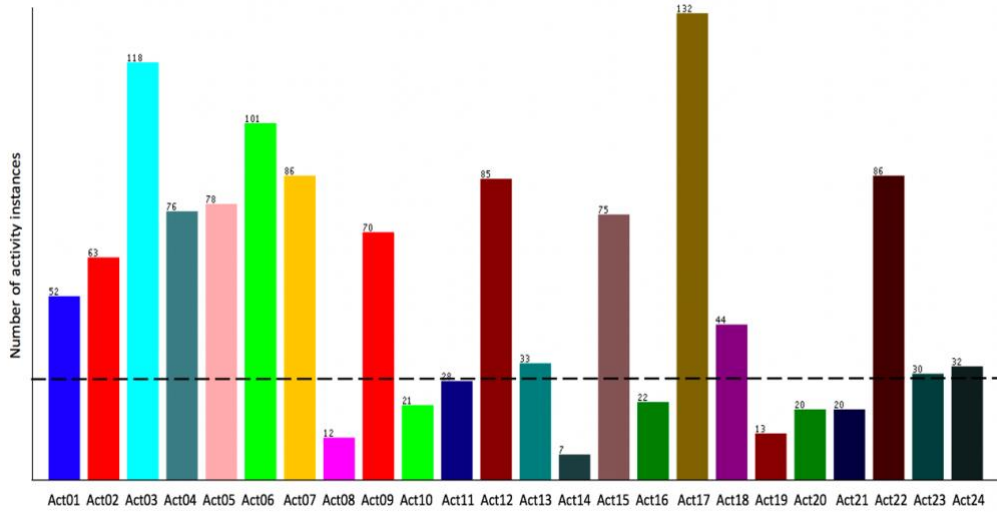


Figure 4.2. Distribution of the 24 UCAMl Cup activity classes with threshold of <30 instances presented as dotted line.

Table 4.5. Activity Classes in the Restructured Dataset, where underrepresented activities in the original dataset have been either removed or merged

ID	Name	Instances	Routine
Act01	Take medication	52	A, E
Act15	Put waste in the bin	75	A, E
Act17	Brush teeth	132	M, A, E
Act18	Use the toilet	44	M, A, E
Act22	Dressing	86	M, A, E
Act23	Go to bed	30	E
Act24	Wake up	32	M
ActN1	Door	61	M, A, E
ActN2	Watch TV on sofa	155	M, A, E
ActN3	Breakfast	141	M
ActN4	Lunch	219	A
ActN5	Dinner	162	E

4.3 Data Pre-Processing

Given that the data restructuring process involved combining the provided train and test sets to produce a set of data that better represented activity classes in the training data, it was subsequently required to extract a new test set. Thus, 15% of the data was randomly selected and removed to generate an unseen test set. The raw data files containing data streams produced by binary sensors included a

timestamp, the sensor ID, the sensor state, and the inhabitant name, as presented in Figure 4.3.

```

TIMESTAMP;OBJECT;STATE;HABITANT
2017/10/31 11:08:55.0;SM4;Movement;Mario
2017/10/31 11:09:11.0;SM4;No movement;Mario
2017/10/31 11:09:31.0;C14;Pressure;Mario
2017/10/31 11:09:31.0;SM4;Movement;Mario
2017/10/31 11:09:32.0;C14;No Pressure;Mario

```

Figure 4.3. Excerpt from a raw binary data file

The data was segmented into 30 second non-overlapping time windows to identify the segments of data that are likely to contain information regarding activities. Time-based windowing involves dividing the entire dataset equally into fixed time segments that can include a varied quantity of data (sensor activations) occurring within each time window [74]. It is a common approach for segmenting data streams collected through environmental sensors, however, no clear consensus exists for choosing the optimal window size for ADL recognition [172], therefore a 30 second window size was chosen as this was the recommended value in the UCAmI Cup challenge. A total of 31 features were included, which consisted of one feature per binary sensor, and an additional time routine feature representing whether the activity had occurred in the morning, afternoon, or evening, to help distinguish between the similar activities previously outlined. For example, as Act23 go to bed and Act24 wake up use the same pressure sensor located in the bed, the inclusion of a time routine feature can help distinguish these activities due to the human nature of habitually waking up in the morning and going to bed in the evening.

4.4 Recommendations

Based upon the findings through conducting investigations within Chapters 3 and 4, various recommendations have been outlined for screening and pre-processing HAR data, which can be used as guidelines to support data collection protocols. Thus, ensuring the collection of good quality data and encouraging data sharing.

Various sources of attribute noise within a time-series accelerometry dataset were identified and handled within Chapter 3, which subsequently led to enhanced performance. In addition to this, Chapter 4 involved collectively identifying data challenges within an environmentally collected sensor-based dataset. In addition to considerations encompassing all sensor-based HAR data, data type-specific issues have been identified, for example the issues identified within Chapter 3 were more specified to time-series data collected through a data collection protocol. Table 4.6 provides recommendations in handling the identified issues.

Table 4.6. Recommendations for pre-processing HAR data

Data type	Data Issue	Recommended Action
All HAR data	Low quantity of data	Avoid classifiers that require large amounts of data such as deep learning methods
	Class imbalance	Apply re-sampling techniques
	Poor distribution of train and test data	Re-distribute the train and test sets, if possible. Alternatively, apply periodic training / model update
	Interclass similarity	Extract discriminative features
	Sensor failure leading to missing values	Imputation or removal of missing data instances
Time-series accelerometry data	Poor/no sensor calibration	Discard data
	Incorrect sensitivity range	Discard data
	Time delays before/after activity recordings	Remove the noisy portion of data
	Irregular data spikes	Remove the noisy portion of data

The low availability of HAR data is a recognised challenge within this domain. Furthermore, some of the obtainable HAR datasets, for example that of the UCAmI Cup, contain low quantities of data which can hinder the performance of data-driven classifiers. Consequently, in this case, it is recommended to avoid classifiers that require large amounts of data for effective performance, for example,

deep learning methods. Another challenge, specifically class imbalance, is a widely-acknowledged and extensively researched HAR challenge that should be addressed during pre-processing to diminish the adverse classification effects. Numerous well-established resampling techniques exist in handling class imbalance [154], [155], [156], thus the application of these techniques is recommended such as undersampling of the majority classes and oversampling of the minority classes to achieve an even distribution of data. Furthermore, the distribution of training and test data is an important aspect for researchers to take into consideration, which will impact upon classification performance. The training set must contain an adequate amount of representative data to train each activity class, otherwise the chosen classifier will be unable to effectively recognise the unseen test data. An example of this scenario was demonstrated within the UCAmI cup challenge, as Act09 watch TV was highly under-represented in the training set, yet the test set included a large number of Act09 instances. Thus, it is recommended to assess the distribution of training and test sets, and redistribute the data, if required, to avoid this issue. Another commonly emerging issue within HAR datasets is that of interclass similarity. This issue hinders classification performance, thus, it is recommended to resolve this issue through extracting more discriminative features. For example, within the UCAmI dataset, Act23 go to bed, and Act24 get up, both involved the same sensor, therefore the inclusion of an additional time feature could discriminate between these activities. Finally, sensor failure has been identified as an issue to consider during pre-processing, which could result in missing values within the dataset and therefore hinder classification performance. It is recommended to handle missing values through either imputation or removal of the data instance, depending on the scenario. For example, if a large amount of data is missing, it could be beneficial to impute data to preserve the dataset, whereas if a large amount of data is available, and only a small amount of data is missing, it is recommended to remove the data instances containing missing values [19].

Considering time-series data similar to that of Chapter 3, various recommendations are outlined to avoid the identified data quality issues and therefore improve classification performance. It is recommended to discard data that has been collected with a poorly calibrated or entirely uncalibrated sensor, or that has been poorly configured through selecting an incorrect sensitivity range. Furthermore, considering data containing only portions of noise, it is recommended to only remove

these noisy portions, thus preserving the remaining good quality, representative activity data. For example, time delays before or after recordings and irregular data spikes within the data should be removed. The data collection protocol should also be carefully reviewed to ensure that users are fully aware of their responsibilities during unsupervised data collection, thus minimising the introduction of noise.

4.5 Conclusion

As stated, data preparation is a vital consideration within machine learning, where a large proportion of time consumption is expended on data collection, cleansing and interpretation [80]. Particularly, generating data-driven classification models require high quality data during training to produce optimally performing and robust models. Yet, the recognised shortage of large, high quality and correctly annotated publicly available datasets continues to delay further advancements in HAR research.

This Chapter assessed the quality of a publicly available HAR dataset and prepared this data for use in succeeding Chapters. A number of challenges were discovered upon initial exploration of the data. Consequently, this Chapter involved improving the quality of an openly available HAR dataset for the purpose of data-driven HAR, as previously recognised in Chapter 3, data quality is a significant consideration whilst exploring data-driven approaches to HAR. The importance of adhering to good data preparation practices was also highlighted, as restructuring the data will support and enhance HAR performance. The data issues discovered in this Chapter may aid the refinement of further data collection protocols. Findings within this Chapter support and reinforce the need for investigations that will aid the development of high quality and refined study protocols, as according to [81], this remains vital in providing publicly available, effective research datasets. Additionally, ascertaining the reliability and quality of pervasive health datasets has been recognised as a challenge, as no clear standards exist in effectively disseminating and re-using these research testbeds [81]. Thus, clear standards need to be developed to ensure the effective collection and sharing of data.

According to [80], another important consideration for completing HAR tasks includes performing feature engineering. Selecting an optimal feature vector is a

crucial, yet time consuming, stage in knowledge discovery [92], which includes the reduction or removal of redundant features as these may cause implications such as needlessly increasing computation time during classification, and adversely affecting classification performance [91]. Furthermore, according to [184] feature engineering can have a substantial influence on the performance of classifiers. Thus, Chapter 5 will investigate the impact of various feature selection techniques for the optimisation of HAR.

Chapter 5

Selecting an Optimal Subset of Features

5.1 Overview

Feature selection is a fundamental stage of the HAR process which involves distinguishing an optimal subset of features required to classify activities most effectively. The discovery and removal of irrelevant features may improve prediction quality and classification performance, whilst also reducing the complexity of the data, computation time and data storage requirements [20].

Common feature selection approaches include the application of filter and wrapper techniques. Many studies conclude that wrappers outperform filter methods in reducing feature dimensionality with accelerometry-based datasets [21]–[25]. Nevertheless, it was recognised that relatively less effort has been made in exploring

the most suitable approach to feature selection with binary datasets within smart environment settings. Thus, a motivating factor for studies conducted in this Chapter is to investigate the impact of both filter and wrapper techniques with binary sensor-based activity data. Furthermore, hybrid approaches for selecting the most effective subset of features have been explored recently. These hybrid methods involve the combination of common feature selection techniques in an attempt to exploit the advantages of each, thus achieving higher classification performance with the resulting feature subset. The opportunity of exploring a hybrid approach to ascertain an optimal subset of features emerged during experiments conducted within this Chapter and was explored. The predominant contribution of this Chapter subsequently involved the development of a new hybrid approach to feature selection.

The remainder of this Chapter is structured as follows: Section 5.2 details the methodology undertaken, Section 5.3 presents the results of initial experimentations conducted, and a hybrid feature selection method is explored in Section 5.4. Finally, Section 5.5 concludes this Chapter.

5.2 Methodology

The dataset utilised in experiments within this Chapter was previously introduced and described in Chapter 4. It comprised of a multi-dimensional dataset generated in a smart apartment, consisting of 31 features and 12 activity classes. These features were largely derived from binary sensor states, previously presented in Table 4.2, in addition to a time routine feature distinguishing whether the activity occurred in the Morning, Afternoon or Evening.

Various feature selection techniques were explored in Weka to discover an optimal subset of the features described, thus redundant or irrelevant features were to be removed in an attempt to increase classification accuracy of the 12 activity classes. The explored techniques included both filter and wrapper methods, which were each evaluated on 4 classifiers, namely kNN, SVM, NN and LR, as this collection of classifiers were evaluated across previous feature selection studies [25], [92], [93], [96]. The 4 classifiers were constructed within Matlab. Matlab recommend various configurations per classifier within the Classification Learner

application, thus each recommended configuration was implemented, and consequently the best performing configuration, per model, was utilised. The chosen evaluation metric for research endeavours conducted within Chapters 5, 6 and 7 is classification accuracy. Due to the HAR dataset utilised within these study chapters deriving from the UCAmI Cup competition, it was decided to conform with their chosen evaluation metric. Classification accuracy is an extensively utilised evaluation metric used in many HAR studies [85], [86], [98], [106], [184]. As described in Chapter 4, an unseen test set was extracted from the restructured dataset. Extracting an unseen test set, also described as holdout data, is a technique utilised to provide an unbiased evaluation during classification, as the performance of the models are evaluated upon data that has not been used during the training phase. Thus, the models were evaluated with 10-fold cross validation, then the unseen test set was subsequently introduced to produce the final classification performance. The same evaluation technique has been utilised in succeeding Chapters.

All filter methods adopted have utilised the ranking search method. With ranking, all features are ordered according to a calculated measure of feature ‘value’ [94]. The wrapper methods have utilised the Best First search method, which involves searching various feature subsets through greedy hillclimbing that is further enhanced with a backtracking capability, which enables this search method to revert back to a previously evaluated subset if the subsequent feature subsets during evaluations do not improve consecutively [191]. Furthermore, the attribute selection mode chosen was cross-validation for both the filter and wrapper techniques.

Sections 5.2.1 to 5.2.4 describe each of the feature selection methods explored during the experiments conducted in this Chapter. These Sections also include the outputs produced following the implementation of each method.

5.2.1 Information Gain

Information Gain is a well-established entropy-based feature selection method, commonly explored in the realms of machine learning to measure the dependence amongst two variables [192]. Implementing this method involves calculating the information gain provided by each attribute individually in relation to the output class, where attributes with higher information gain values provide more

information and are therefore more relevant. In terms of classification, the number of times each feature occurs per category are counted to calculate the information gain contributed per feature [192]. Each feature is ranked in order of importance (determined by the level of information provided), thus the least relevant features can be removed.

The average merit values produced by this method range from 0 to 1, with a score of 0 indicating that no information was obtained from the feature and a score of 1 indicating that maximum information was obtained. Figure 5.1 presents the outputs produced by the Information Gain filter when evaluated on the considered dataset.

	average merit	average rank	attribute
	0.931 +- 0.005	1 +- 0	31 Routine
	0.541 +- 0.007	2 +- 0	25 SM1
	0.291 +- 0.005	3.1 +- 0.3	27 SM4
	0.267 +- 0.009	3.9 +- 0.3	28 SM5
	0.198 +- 0.004	5.1 +- 0.3	26 SM3
	0.184 +- 0.007	6.3 +- 0.64	24 M01
	0.18 +- 0.005	6.6 +- 0.49	29 S09
	0.157 +- 0.006	8.1 +- 0.3	16 D03
	0.148 +- 0.005	8.9 +- 0.3	8 C09
	0.112 +- 0.005	10.5 +- 0.81	12 C14
	0.106 +- 0.005	11.1 +- 0.7	19 D07
	0.105 +- 0.007	11.4 +- 0.66	11 C13
	0.066 +- 0.005	14.3 +- 1.19	9 C10
	0.064 +- 0.005	14.5 +- 1.36	7 C08
	0.063 +- 0.003	14.9 +- 1.45	22 D10
	0.061 +- 0.004	15.4 +- 1.5	17 D04
	0.059 +- 0.003	15.9 +- 0.83	1 C01
	0.046 +- 0.003	18.3 +- 0.46	5 C05
	0.042 +- 0.003	19.1 +- 0.7	15 D02
	0.041 +- 0.003	19.7 +- 0.78	14 D01
	0.033 +- 0.004	21.5 +- 0.67	30 TV0
	0.033 +- 0.002	21.5 +- 0.67	4 C04
	0.03 +- 0.002	22.9 +- 0.3	20 D08
	0.015 +- 0.002	24 +- 0	23 H01
	0.011 +- 0.002	25.2 +- 0.4	10 C12
	0.01 +- 0.001	25.8 +- 0.4	18 D05
	0.003 +- 0.001	27 +- 0	6 C07
	0 +- 0	28.2 +- 0.6	2 C02
	0 +- 0	29.4 +- 0.49	3 C03
	0 +- 0	29.9 +- 1.14	13 C15
	0 +- 0	30.5 +- 0.5	21 D09

Figure 5.1. Output values produced by Information Gain

5.2.2 Correlation

The correlation-based method of selecting optimal features assesses the relevance of each by observing intercorrelation between features as well as examining their ability to predict the output class [86]. The optimal subset of features

chosen through correlation include those that present high correlation with the output class and no intercorrelation with other attributes in the feature space. Features that are highly intercorrelated may demonstrate the problem of multicollinearity, in that if two or more features are too closely related to one another, the performance of some classification models can diminish due to excessive complexity, thus resulting in unstable models [193].

The Correlation filter produces average merit values ranging from -1 to 1. A score of -1 indicates negative correlation whereas a score of 1 indicates positive correlation. Additionally, features with no correlation to the target class will produce a merit value of 0. Figure 5.2 presents the average merit values produced by the Correlation filter when evaluated on the considered dataset.

average merit	average rank	attribute
0.267 +- 0.002	1 +- 0	25 SM1
0.239 +- 0.001	2 +- 0	31 Routine
0.183 +- 0.003	3.2 +- 0.4	28 SM5
0.18 +- 0.001	3.8 +- 0.4	27 SM4
0.153 +- 0.001	5.1 +- 0.3	26 SM3
0.152 +- 0.002	5.9 +- 0.3	29 S09
0.132 +- 0.002	7 +- 0	8 C09
0.121 +- 0.002	8.1 +- 0.3	24 M01
0.115 +- 0.002	8.9 +- 0.3	16 D03
0.096 +- 0.003	10 +- 0	11 C13
0.088 +- 0.002	11 +- 0	22 D10
0.075 +- 0.002	12.9 +- 1.04	19 D07
0.075 +- 0.001	13.2 +- 0.87	12 C14
0.073 +- 0.003	14.4 +- 2.33	7 C08
0.071 +- 0.002	15.2 +- 1.47	4 C04
0.068 +- 0.003	17.2 +- 2.09	17 D04
0.068 +- 0.002	17.3 +- 1.55	20 D08
0.068 +- 0.004	17.9 +- 2.66	15 D02
0.068 +- 0.002	18.1 +- 1.51	14 D01
0.066 +- 0.004	18.5 +- 1.63	30 TV0
0.062 +- 0.002	21.1 +- 0.83	9 C10
0.062 +- 0.002	21.2 +- 0.98	1 C01
0.055 +- 0.002	23 +- 0	5 C05
0.044 +- 0.002	24 +- 0	23 H01
0.036 +- 0.003	25.2 +- 0.4	18 D05
0.032 +- 0.003	25.8 +- 0.4	10 C12
0.019 +- 0.006	27 +- 0	6 C07
0 +- 0	28.4 +- 0.8	2 C02
0 +- 0	29 +- 0	3 C03
0 +- 0	30.2 +- 0.4	13 C15
0 +- 0	30.4 +- 1.2	21 D09

Figure 5.2. Output values produced by Correlation

5.2.3 Relief-F

The Relief-F method of selecting features is an extended form of the Relief algorithm developed to deal with multiclass problems [192]. The Relief algorithm

ranks features by applying weights to each based upon the correlation between each feature and class. According to [94], filtering algorithms based on variations of Relief, such as Relief-F, are the only filters possessing the ability to detect dependencies of features indirectly through their adoption of the nearest neighbour concept. This method discovers the nearest “hits” as data observations belonging to the same class, and “misses” as data observations belonging to different classes, rather than directly and exhaustively searching through countless feature combinations [192]. Subsequently, the features are ranked based upon their relevance which is determined by how well data observations from the same class, and those from different classes, are distinguished. Figure 5.3 presents the concept of nearest neighbours in the Relief-F algorithm, with the number of neighbours set to 3 for illustrative simplicity. It can be seen that there are 3 nearest “hits” as well as 3 nearest “misses” in the feature space that are highlighted in relation to the target class.

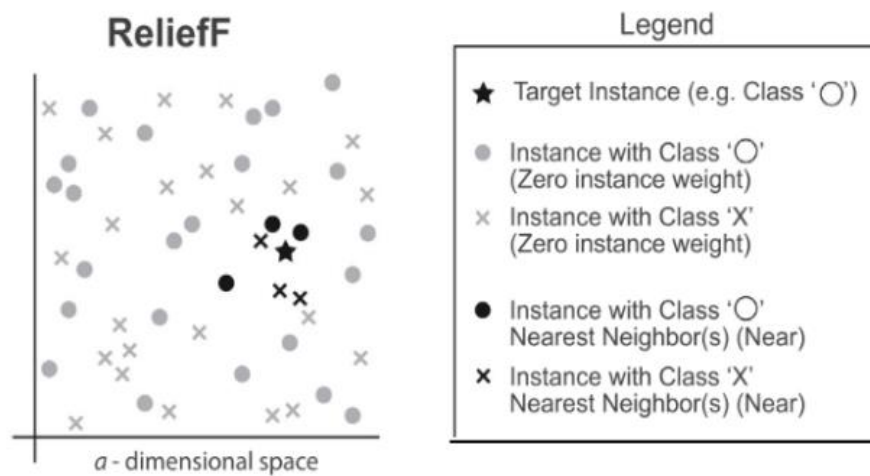


Figure 5.3. Relief-F concept illustrating the neighbour selection [94]

The main parameter to consider with this filtering algorithm is setting the number of neighbours. During the establishment of the Relief-F variation, initial empirical testing of the algorithm recommended the use of 10 nearest neighbours which has since been recognised as the default setting due to its extensive application [94]. Relief-F is capable of assigning different weights per instance using a distance-based measure in relation to the target. This measure is known in Weka as the Sigma value, which regulates the pace at which weights decrease for instances situated

further in distance from the target. The recommended Sigma value to adopt is stated as being between 1/5 to 1/10 of the quantity of nearest neighbours specified, with the default value provided set as 2. Due to the reviewed literature, the default parameters for Relief-F were chosen for experimentation.

The Relief-F filter produces average merit values for each feature ranging from 1 to -1, calculated through summing the weighted differences in the same class and different classes. A higher merit value indicates that the feature being assessed is differentially conveyed, meaning greater differences are expressed for data instances from different classes. Figure 5.4 presents the outputs produced by the Relief-F filter when evaluated on the considered dataset.

	average merit	average rank	attribute
Valuable features	0.408 +- 0.005	1 +- 0	31 Routine
	0.367 +- 0.017	2 +- 0	25 SM1
	0.165 +- 0.007	3 +- 0	27 SM4
	0.136 +- 0.009	4 +- 0	28 SM5
	0.088 +- 0.012	5.4 +- 0.66	29 S09
	0.078 +- 0.007	5.8 +- 0.4	26 SM3
	0.069 +- 0.006	6.9 +- 0.7	24 M01
	0.053 +- 0.006	8.4 +- 0.66	8 C09
	0.053 +- 0.004	8.5 +- 0.5	16 D03
	0.031 +- 0.003	10.6 +- 0.49	12 C14
	0.031 +- 0.008	10.9 +- 1.14	22 D10
	0.024 +- 0.003	12.4 +- 1.2	11 C13
	0.022 +- 0.002	12.7 +- 0.78	19 D07
	0.016 +- 0.002	14.9 +- 0.94	1 C01
	0.014 +- 0.007	16.3 +- 3.58	14 D01
	0.014 +- 0.003	17 +- 1.48	17 D04
	0.013 +- 0.003	17 +- 1.48	7 C08
	0.011 +- 0.004	17.8 +- 1.72	15 D02
	0.011 +- 0.005	18.3 +- 2.76	30 TV0
	Irrelevant features	0.009 +- 0.003	19.9 +- 1.87
0.008 +- 0.002		19.9 +- 1.7	9 C10
0.007 +- 0.004		21 +- 2.32	20 D08
0.004 +- 0.002		22.3 +- 1	5 C05
0.001 +- 0		24 +- 0	23 H01
0 +- 0		26.1 +- 2.21	18 D05
0 +- 0		27 +- 1.79	6 C07
0 +- 0		27.6 +- 1.5	3 C03
0 +- 0		27.7 +- 0.46	2 C02
0 +- 0		28 +- 1.34	21 D09
0 +- 0	29.2 +- 0.6	13 C15	
-0 +- 0	30.4 +- 1.8	10 C12	

Figure 5.4. Output values produced by Relief-F

5.2.4 Sequential Selection: Forward and Backward

The two wrapper approaches used were sequential forward selection (SFS) and sequential backward selection (SBS). SFS generates an optimal subset of features based on the chosen classifier by incrementally adding features to the selection. The process begins with an empty selection, which develops by increasing

this selection by one feature per round. During each round, per each added feature, the model performance is evaluated where the feature contributing to the best performance is retained and added to the optimal subset [95]. SBS operates in the reverse direction to SFS in that the process begins with the full feature vector, with the selection of features subsequently decreasing by one feature per round. As with the previous approach, the model performance is evaluated per round, per each removed feature, where the feature providing the least diminishing performance is removed from the final chosen subset [95].

Unlike the filter methods, there are no rankings produced by the wrapper methods. Instead, the output produced by wrapper methods present the number of folds in which each feature was selected for retention in the dataset, as the attribute selection mode was cross-validation. Figure 5.5 presents an example of the output produced by the kNN wrapper method in conjunction with the SFS search technique.

number of folds (%)	attribute
10(100 %)	1 C01
0(0 %)	2 C02
0(0 %)	3 C03
9(90 %)	4 C04
9(90 %)	5 C05
0(0 %)	6 C07
10(100 %)	7 C08
10(100 %)	8 C09
2(20 %)	9 C10
0(0 %)	10 C12
10(100 %)	11 C13
2(20 %)	12 C14
0(0 %)	13 C15
10(100 %)	14 D01
10(100 %)	15 D02
10(100 %)	16 D03
9(90 %)	17 D04
8(80 %)	18 D05
10(100 %)	19 D07
10(100 %)	20 D08
0(0 %)	21 D09
10(100 %)	22 D10
0(0 %)	23 H01
10(100 %)	24 M01
10(100 %)	25 SM1
10(100 %)	26 SM3
2(20 %)	27 SM4
10(100 %)	28 SM5
10(100 %)	29 S09
5(50 %)	30 TV0
10(100 %)	31 Routine

Figure 5.5. Output produced by Wrapper method

5.2.5 Rationale for Feature Selection Threshold

According to [94], no clear consensus exists as to which features should be retained/removed from the feature space. Instead, they recommend the implementation of an ad-hoc threshold which can be ascertained by either a statistical or subjective likelihood of feature significance, or merely a preferred quantity of features to be retained/removed from the final chosen subset.

As stated by [94], the removal of presumably irrelevant or redundant features should be performed with caution, as they may still provide useful information through their inclusion. This may be the case with certain features in the considered dataset. The merit values were very low for many of the ranked features. Thus, based upon the aforementioned literature, and also through analysing the dataset and ranked feature merits with subjective likelihood in relation to identified primary sensors required to recognise certain activities, it was intuitively decided to employ a cut-off threshold of 0. Due to this threshold decision, as many useful features as possible were retained, as caution was taken to avoid the removal of any potentially valuable sensor features.

Table 5.1 presents the features retained and removed for each feature selection method based upon this threshold. Considering all feature selection methods collectively, there were 23 features chosen for retention within the feature space in all scenarios, whereas the remaining 8 features were deemed irrelevant by one or more method. These features included: C02 fruit platter, C03 cutlery, C07 XBOX remote, C12 laundry basket, C15 kitchen faucet, D05 dishwasher, D09 washing machine and H01 kettle. A clear distinction of the 4 least relevant features emerged, as C02 fruit platter, C03 cutlery, C15 kitchen faucet and D09 washing machine were only chosen for retention by 1 or 2 methods each. Furthermore, the Relief-F filter and the kNN SFS wrapper had chosen to remove the most features, with each removing 7 from the feature space.

Considering only the filter methods, namely Information Gain, Correlation and Relief-F, it was seen that features C02 fruit platter, C03 cutlery, C15 kitchen faucet, and D09 washing machine were deemed irrelevant as these features were never chosen for retention with any filter methods. As for feature H01 kettle, it was recognised that all filter methods had chosen to retain this, thus supporting its

relevance. Notably, features C07 XBOX remote, C12 laundry basket, and D05 dishwasher were chosen for removal by the Relief-F algorithm, whereas the Information Gain and Correlation filters had retained these features.

Considering only the wrapper methods, more variation occurred in which methods chose to retain or remove certain features. Features H01 kettle and C07 XBOX remote were equal in that 3 wrapper methods chose to retain, and the remaining 5 wrapper methods chose to remove them. As for features C02 fruit platter, C03 cutlery, and D09 washing machine, these were only chosen for retention twice each, and C15 kitchen faucet was only chosen for retention once. Finally, considering D05 dishwasher, and C12 laundry basket, all wrapper methods had chosen to retain these features apart from 1 method each, namely the SFS NN method in relation to D05 dishwasher and the SFS kNN method in relation to C12 laundry basket.

Table 5.1. Features considered in each experiment, where Y indicates inclusion in the subset and N indicates removal of the feature

	Filters			Wrappers								
	Ranking search method			SFS Best First search method				SBS Best First search method				
Feature	Information Gain	Correlation	Relief-F	kNN	SVM	NN	LR	kNN	SVM	NN	LR	Total
C01 medication box	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
C02 fruit platter	N	N	N	N	N	Y	N	N	N	Y	N	2
C03 cutlery	N	N	N	N	N	Y	N	N	N	Y	N	2
C04 pots	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
C05 water bottle	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
C07 XBOX remote	Y	Y	N	N	N	Y	N	N	Y	Y	N	5
C08 trash	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
C09 tap	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
C10 tank	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
C12 laundry basket	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	9
C13 pyjamas drawer	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
C14 bed	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
C15 kitchen faucet	N	N	N	N	N	Y	N	N	N	N	N	1
D01 refrigerator	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
D02 microwave	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
D03 wardrobe	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
D04 cups cupboard	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
D05 dishwasher	Y	Y	N	Y	Y	N	Y	Y	Y	Y	Y	9
D07 WC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
D08 closet	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
D09 washing machine	N	N	N	N	N	Y	N	N	N	Y	N	2
D10 pantry	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
H01 kettle	Y	Y	Y	N	N	Y	N	N	N	Y	Y	6
M01 door	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
SM1 kitchen area	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
SM3 bathroom area	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
SM4 bedroom area	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
SM5 sofa area	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
S09 sofa	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
TV0 TV	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
Time Routine	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	11
Total Features	27	27	24	24	25	30	25	25	26	30	26	

5.3 Initial Experimental Results

Table 5.2 provides a baseline for experiments conducted in this Chapter, presenting the classification accuracies in which the suite of classifiers have recognised activities based upon the full original feature vector, where no feature selection has been applied. The LR classifier performed best during the testing phase with the full feature vector, whereas the kNN model performed least effectively.

Table 5.2. No feature selection applied. All original 31 features are included.

Exp.	Classifiers	Train Accuracy (%)	Test Accuracy (%)
All Features	kNN	75.10	70.95
	SVM	77.90	76.54
	NN	83.47	80.45
	LR	82.57	81.01

Table 5.3 presents the results of the wrapper approaches applied to the binary dataset. The LR model performed best with both wrapper methods, namely SFS and SBS, whereas the kNN classifier performed least effectively across both methods. In comparison to Table 5.2, the LR model increased both train and test performances with the SFS and SBS methods, whilst also reducing the number of features by 6 and 5, respectively. Considering the kNN model, a larger number of features were removed, specifically 7 and 6 per method, respectively, whilst also maintaining test performance and exhibiting slight variation in train performance. The SVM model demonstrated more variation whilst generating comparisons between the SFS and SBS methods. Considering the SFS method with the removal of 6 features, the SVM training performance decreased whilst presenting the ability to maintain performance on the unseen test data. Comparatively, reflecting upon the SBS method with the removal of 5 features, the SVM training performance increased, whilst the test performance decreased slightly. Contrarily, only 1 feature was removed with both wrapper methods on the NN model, with classification accuracies decreasing slightly with the training set and increasing when evaluated on the unseen test set.

Based upon this analysis, the SFS method was deemed most effective in comparison to the SBS method in terms of reducing data dimensionality slightly more, whilst either maintaining or enhancing the unseen test performances across the suite of classifiers. The training performances had decreased slightly across more of the classifiers with SFS, in comparison to demonstrating more increases with SBS, nevertheless, the SFS models had proven their enhanced stability on the test data.

Table 5.3. Feature selection applied via Wrapper techniques, where the values in brackets indicate the comparative difference between the current performance and the performances achieved with no feature selection applied (Table 5.2)

Exp.	Classifiers	No. of Features Removed	Train Accuracy (%)	Test Accuracy (%)
Sequential Forward Search (SFS)	kNN	7	75.00 (-0.10)	70.95
	SVM	6	77.40 (-0.50)	76.54
	NN	1	83.17 (-0.30)	81.01 (+0.56)
	LR	6	83.56 (+0.99)	82.12 (+1.11)
Sequential Backward Search (SBS)	kNN	6	75.20 (+0.10)	70.95
	SVM	5	78.50 (+0.60)	76.34 (- 0.20)
	NN	1	83.07 (-0.40)	81.01 (+0.56)
	LR	5	83.56 (+0.99)	82.12 (+1.11)

Table 5.4 presents the results of the filter approaches applied to the binary dataset. The LR model performed best across all methods, whereas the kNN model performed least effectively. Whilst generating comparisons to Table 5.2 (the full feature vector), the entire suite of classifiers either maintained or enhanced their classification performance on the unseen test data across all filter methods. A total of 4 features were removed with the Information Gain filter, with results exhibiting improved performance on 3 out of 4 classifiers during training, whilst maintaining performance with the remaining classifier, namely the NN model. As for the test data, the filtered feature vector derived through Information Gain improved the performance of 2 classifiers whilst maintaining performance for the remaining models. Considering the Correlation filter, 4 features were removed following its implementation. Results demonstrated enhanced performance on 3 models during

training and 2 models during the test phase, with the remaining models maintaining their performance. As for the Relief-F algorithm, a total of 7 features were removed. During training, 2 classifiers improved their performance, whilst the NN model decreased, and the remaining model, namely kNN, was able to maintain its performance. Additionally, during the testing phase all models had improved their classification performance.

Based upon this analysis, Relief-F was deemed the most effective filtering method as the highest number of features were removed as well as all 4 classifiers improving their performance on the unseen test set. Comparatively, the Information Gain and Correlation methods improved their performance on the unseen test data on 2 out of 4 classifiers, with less features removed.

Table 5.4. Feature selection applied via Filter techniques, where the values in brackets indicate the comparative difference between the current performance and the performances achieved with no feature selection applied (Table 5.2)

Exp.	Classifiers	No. of Features Removed	Train Accuracy (%)	Test Accuracy (%)
Information Gain	kNN	4	75.60 (+0.50)	70.95
	SVM		78.60 (+0.70)	76.54
	NN		83.47	81.56 (+1.11)
	LR		83.56 (+0.99)	82.57 (+1.56)
Correlation	kNN	4	75.60 (+0.50)	70.95
	SVM		78.60 (+0.70)	76.54
	NN		83.47	81.56 (+1.11)
	LR		83.56 (+0.99)	82.57 (+1.56)
Relief-F	kNN	7	75.10	71.50 (+0.55)
	SVM		78.10 (+0.20)	77.10 (+0.56)
	NN		82.87 (-0.60)	81.01 (+0.56)
	LR		82.97 (+0.40)	81.56 (+0.55)

Considering both wrapper and filter approaches collectively, it was concluded that the filter methods outperformed the wrappers. Through analysing Tables 5.3 and 5.4, the best performing wrapper was deemed the SFS method and the best performing filter was deemed the Relief-F algorithm. Thus, comparisons were ultimately generated upon these approaches. As stated, the Relief-F filter removed a

total of 7 features, whereas the SFS method varied between removing 1-7 features across all classifiers. Additionally, the training performances based on the Relief-F algorithm increased with 2 models and decreased with the NN, whereas with the SFS wrapper method the training performances increased only with the LR model and decreased with 3 models. The testing performances based on the Relief-F algorithm increased with all 4 classification models, whereas with the SFS wrapped method, only 2 models achieved an increase in test performance.

In summary, the benefits of performing feature selection, and thus reducing data dimensionality, were demonstrated across all methods evaluated. The obtained experimental results indicated that redundant features may be removed from the dataset without hindering classification performance. Instead, the performances may be either maintained or improved with less features included.

5.4 Hybrid-Filter Approach

As previously stated, hybrid approaches to feature selection have been explored more recently in several research domains due to their perceived benefits. The combination of established methods aims to exploit the positive characteristics of each, whilst potentially diminishing their limitations.

Section 5.4.1 provides the rationale for the proposed hybrid-filter approach, Section 5.4.2 describes the methodology undertaken and Section 5.4.3 presents the experimental results obtained through implementation of the proposed approach and also provides a comprehensive discussion of results.

5.4.1 Rationale

Since examining the features removed through filtering approaches collectively, previously presented in Table 5.4, it was realised that each filter method had removed 4 common features, namely C02 fruit platter, C03 cutlery, C15 kitchen faucet, and D09 washing machine, with the only differences between Information Gain and Correlation, and the Relief-F algorithm being that Relief-F had removed an additional 3 features, namely D05 dishwasher, C12 laundry basket and C07 XBOX

remote. Furthermore, the classification performance had decreased slightly on some classifiers with Relief-F in comparison to the other filtering approaches. Thus, through deliberating over whether the 7 features removed through Relief-F were truly optimal, it was decided to combine the filters to generate new hybrid feature subsets.

According to [192], combining filter methods demonstrated benefits over their individual forms as the advantages of each method were heightened through positively complementing one another. In [192], a hybrid-filter method was proposed through combining Information Gain and Relief-F. Conclusions stated that the proposed hybrid-filter was able to prevent the Relief-F algorithm from neglecting the underrepresented minority classes by enhancing feature impact on those, whilst preserving the impact of features on classes with a larger number of instances, specifically the overrepresented majority classes. Other cited benefits of combining these filters included the avoidance of excessively irrelevant or redundant features, the efficient reduction of valuable information loss, and finally, the reduction in time consumption required in comparison to wrapper methods.

Based upon this analysis, the opportunity emerged to combine the feature vectors from different filter approaches, and therefore the potential of a hybrid-filter method was explored, thus a main contribution of this Chapter involved the development of a new hybrid feature selection method that produced an optimal subset of features.

5.4.2 Methodology

The first stage of combining filter methods involved defining a base set of features to be removed. These base features were derived through identifying the common features chosen for removal from each filter method through exploration of the AND operator, as these were deemed extensively redundant. Subsequently, the remaining features were organised upon the XOR operator which represented features chosen for removal by one filter method, and not chosen by the other method, as these were deemed seemingly redundant whilst also possessing the potential of providing valuable information. As a result, two feature pools were established, presented in Figure 5.6, in which the AND features comprised of C02 fruit platter,

C03 cutlery, C15 kitchen faucet, and D09 washing machine, and the XOR features comprised of D05 dishwasher, C07 XBOX remote and C12 laundry basket.

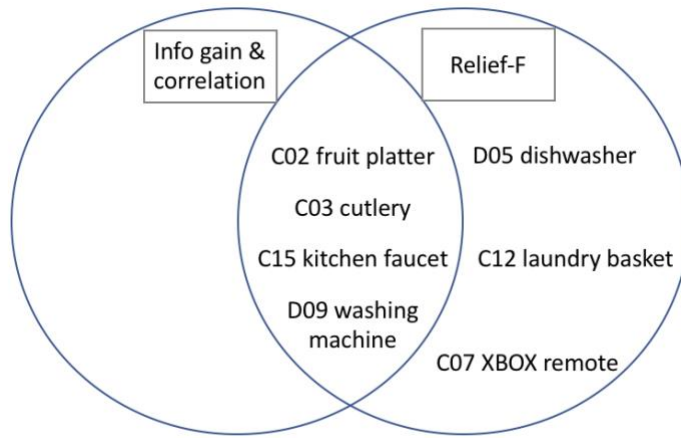


Figure 5.6. Filter feature pools demonstrating the AND and XOR features

Based upon the generated feature pools, 6 new feature subsets were derived. All newly formed subsets initially involved the removal of the 4 base features, namely C02 fruit platter, C03 cutlery, C15 kitchen faucet and D09 washing machine. Further to this, an exhaustive search method was implemented to evaluate the ascertained XOR features, namely D05 dishwasher, C12 laundry basket and C07 XBOX remote, in combination with the base set to explore all hybrid-filter subset eventualities, thus potentially discovering an optimal combination of features. Table 5.5 presents the hybrid-filter feature subsets.

Table 5.5. Newly generated feature subsets based upon combined filters

Subset #	Features removed
1	(-5) 4 base features + C07
2	(-5) 4 base features + C12
3	(-5) 4 base features + D05
4	(-6) 4 base features + C07 & C12
5	(-6) 4 base features + C07 & D05
6	(-6) 4 base features + C12 & D05

The feature subsets presented in Table 5.5 were then evaluated on the suite of classifiers defined with the previously evaluated feature selection methods to generate performance comparisons.

5.4.3 Results and Discussion

Table 5.6 presents the results of the proposed hybrid-filter method applied to the binary dataset, where values presented in brackets indicate comparisons to results presented in Table 5.2 with no feature selection applied. Whilst considering Subsets 1-3, each had removed a total of 5 features, whereas Subsets 4-6 had each removed 6 features. During training, all hybrid feature subsets had increased performance with all 4 classification models. As for the test performances, Subsets 1, 2, and 4 had demonstrated an increase in accuracies on 2 classifiers, with the remaining models maintaining their performance. Contrarily, Subsets 3, 5 and 6 had exhibited an increase in accuracies with all 4 classifiers. Of these, Subsets 5 and 6 had removed an additional feature in comparison to Subset 3, thus indicating their further effectiveness. Subset 6 had shown larger increases in classification accuracy compared to those in Subset 5, therefore it was deemed the most effective hybrid approach.

In comparison to Table 5.4 comprising the individual filter results, many of the classification performances with the newly generated hybrid subsets had improved, with additional features also removed. Subsets 3, 5 and 6 outperformed both Information Gain and Correlation in terms of more features being removed in addition to achieving enhanced performance with all 4 classifiers on the test set. Subset 6 of hybrid filtering also outperformed the Information Gain and Correlation methods in terms of achieving higher classification accuracies on 3 of the 4 classifiers with regards to both training and test sets. Overall, the best performing hybrid filtering approach was Subset 6.

Table 5.6. Classification performances based on newly generated feature subsets, where the values in brackets indicate the comparative difference between the current performance and the performances achieved with no feature selection applied (Table 5.2)

Exp.	Classifiers	No. of Features Removed	Train Accuracy (%)	Test Accuracy (%)
Subset 1	kNN	5	75.35 (+0.25)	70.95
	SVM		78.32 (+0.42)	76.54
	NN		84.87 (+1.40)	82.68 (+2.23)
	LR		83.56 (+0.99)	82.12 (+1.11)
Subset 2	kNN	5	75.25 (+0.15)	70.95
	SVM		78.61 (+0.71)	76.54
	NN		86.00 (+2.53)	82.68 (+2.23)
	LR		83.66 (+1.09)	82.12 (+1.11)
Subset 3	kNN	5	75.45 (+0.35)	71.50 (+0.55)
	SVM		78.61 (+0.71)	77.10 (+0.56)
	NN		85.15 (+1.68)	82.12 (+1.67)
	LR		83.66 (+1.09)	82.12 (+1.11)
Subset 4	kNN	6	75.35 (+0.25)	70.95
	SVM		78.12 (+0.22)	76.54
	NN		85.01 (+1.54)	83.24 (+2.79)
	LR		83.66 (+1.09)	82.12 (+1.11)
Subset 5	kNN	6	74.85 (-0.25)	71.51 (+0.56)
	SVM		78.42 (+0.52)	77.10 (+0.56)
	NN		86.00 (+2.53)	81.56 (+1.11)
	LR		83.66 (+1.09)	82.12 (+1.11)
Subset 6	kNN	6	75.54 (+0.44)	71.51 (+0.56)
	SVM		79.01 (+1.11)	77.10 (+0.56)
	NN		84.87 (+1.40)	83.24 (+2.79)
	LR		83.76 (+1.19)	82.12 (+1.11)

Comparisons were then made between the Relief-F algorithm and Subset 6, presented in Table 5.7. Previously, Relief-F was distinguished as the best performing filter method as all 4 classifiers improved accuracies on the test set, having also removed 7 features. Comparatively, 6 features were removed with Subset 6, however a considerable trade-off existed between the number of features removed and the accuracies achieved with the optimal feature vector removed in Subset 6. Considering the training performances, Relief-F improved with 2 classifiers and decreased performance with 1 model, whereas Subset 6 improved with all 4 classifiers. Furthermore, the test set performances for both approaches included improvements on all 4 classifiers. The accuracy values with Subset 6, however, outperformed those of the Relief-F algorithm whilst considering both the train and test sets, thus demonstrating its effectiveness.

Table 5.7. Comparison of all features, Relief-F and Subset 6

Exp.	Classifiers	No. of Features Removed	Train Accuracy (%)	Test Accuracy (%)
All features	kNN	0	75.10	70.95
	SVM		77.90	76.54
	NN		83.47	80.45
	LR		82.57	81.01
Relief-F	kNN	7	75.10	71.50
	SVM		78.10	77.10
	NN		82.87	81.01
	LR		82.97	81.56
Subset 6	kNN	6	75.54	71.51
	SVM		79.01	77.10
	NN		84.87	83.24
	LR		83.76	82.12

Considering knowledge of the sensor types, the results provided by the feature selection methods were consistent. The 4 common features for removal, namely C02 fruit platter, C03 cutlery, C15 kitchen faucet and D09 washing machine were not deemed integral to related activities. For example, C02 fruit platter may rarely trigger within the ActN3 Breakfast, ActN4 Lunch and ActN5 Dinner activities, and therefore may provide minimal information. Furthermore, the D09 washing machine sensor may provide minimal information in recognising the Act22 Dressing activity, as an inhabitant may only rarely perform this activity and subsequently place their clothes in the washing machine. As for the XOR features, namely D05 dishwasher, C12 laundry basket and C07 XBOX remote, knowledge of the activities would indicate that D05 dishwasher would be most redundant given that the ActN3 Breakfast, ActN4 Lunch and ActN5 Dinner activities may not involve the inhabitant washing their dishes, or perhaps they manually wash their dishes. Considering the C12 laundry basket sensor in relation to the Act22 Dressing activity, knowledge would suggest that this sensor would be largely involved, however, it was deemed irrelevant through feature selection. Finally, the C07 XBOX remote sensor could be deemed relevant in relation to the ActN2 Watch TV on Sofa activity as this remote can control the TV, however, it may not be essential as the TV0 sensor is identified as the integral sensor required to perform this activity.

5.5 Conclusion

As presented, the detection and removal of redundant features is an important consideration due to their possible effects on predictive quality and classification accuracy. Reducing data dimensionality also reduces the complexity of the data, computational capacities required, and time consumption expended on computation.

In this Chapter, a number of well-established feature selection techniques were evaluated on a binary sensor-based HAR dataset, which included exploration upon both filter and wrapper methods. Initial experimentation revealed better classification performances with filter methods in comparison to the wrapper techniques, with the Relief-F filter outperforming all other methods in terms of the largest number of redundant features identified and removed, as well as the attainment of enhanced classification accuracies. Subsequently, the opportunity of implementing a hybrid-filter approach was investigated due to the perceived benefits of combining well-established feature selection techniques. Thus, further experiments were conducted with the proposed hybrid-filter method, where newly generated feature subsets were derived and evaluated. Comparisons were then made between the performance achieved through implementing the hybrid method and the original Relief-F filter, where a considerable trade-off existed between the number of features removed and the accuracies achieved.

Since conducting experiments in this Chapter, the benefits of performing feature selection were demonstrated. It was observed that reducing the dimensionality of the data, through evaluating performance with the exclusion of redundant features, lead to the classifiers either maintaining their performance or achieving a positive influence on their predictive quality and classification accuracy. The benefits of combining feature selection methods was also demonstrated through implementation of the proposed hybrid-filter approach, where the combined methods complemented one another to ultimately achieve an optimal subset of features, and therefore classification performance was further improved.

The benefit of combining techniques extend into the classification stage of the HAR process, in which combining classifiers through ensemble methods have been explored recently due to their perceived effectiveness in enhancing classification performance [48]. Rather than exclusively depending upon the performance of one classifier, generating an ensemble method comprising of multiple

models may compensate for the recognised limitations of single models through successful combination methods [50]. Thus, Chapter 6 will investigate the potential of ensemble methods in an endeavour to optimise HAR performance. Notably, the findings within Chapter 5, specifically the optimally selected feature subset, has not been utilised within succeeding Chapters. Instead, research endeavours conducted within succeeding Chapters have utilised the original full feature set.

Chapter 6

Homogeneous Neural Network Ensemble for Human Activity Recognition

6.1 Overview

Ensemble methods have acquired considerable research interest recently due to their ability to improve the performance of classification models [48]. According to [50], the fundamental aim of enhancing generalization capabilities exists as the primary motivation to explore ensemble methods. A motivating factor for studies conducted in this Chapter is to investigate the efficiency of complex NNs by

exploring ensemble learning for sensor-based HAR. As outlined in Section 2.5.3, ensemble generation and integration are two important considerations whilst exploring ensemble methods, thus experimentation within this Chapter involved investigations into both aspects. A new ensemble of NNs is proposed, in addition to exploring various approaches to resolving conflicts that occur between base models within ensembles. Specifically, studies involved data gleaned through binary sensors that have been deployed within a smart environment.

The remainder of this Chapter is structured as follows: Section 6.2 provides the rationale for implementing a homogeneous NN ensemble method, Section 6.3 describes the materials and methods implemented, including the proposed HAR classification model and model conflict resolution. Finally, Section 6.4 presents the results and discussion, and Section 6.5 concludes this Chapter.

6.2 Rationale for Homogeneous NN Ensemble

Ensemble learning for HAR has been explored within this Chapter due to its perceived benefits, such as its ability to enhance classification performance in addition to improving generalisation capabilities. According to [177], NNs are a popular base model choice in generating homogeneous ensembles, and for HAR tasks in particular, due to their ability to learn complex, non-linear decision boundaries [47], thus supporting the decision to implement a homogeneous ensemble of NNs within this Chapter. An ensemble of NNs were explored, though due to a lack of high-quality data in ADL datasets, and the low quantity of available data, it was decided to employ lighter weight models rather than exploring deeper architectures. The literature has suggested that shallow NNs have previously achieved similar performance to deep NN architectures for HAR tasks, with provided recommendations to use shallow architectures particularly in cases where a small number of training samples are available [44], [128]. As stated in [194], one of the crucial problems to consider with ensemble learning is the combination rule employed to determine a final class decision amongst the base models. In this work, a support function integration method was used to fuse the base models, and various approaches to effectively resolve conflicts that occur between the base models were investigated to determine a final output decision.

6.3 Methodology

The materials and methods implemented are described within this Section, the proposed ensemble approach is presented in Section 6.3.1, and conflict resolution techniques are described in Section 6.3.2.

6.3.1 Proposed HAR Classification Model

Recently, ensemble methods for classification tasks have been explored due to their potential to enhance robustness, improve performance and also increase generalisation capabilities in comparison to single model methods [128]. The proposed classification method within this study comprised of combining several NN base classifiers to generate a homogeneous ensemble. The UCAmI cup dataset was utilised for experiments conducted within this Chapter, which was introduced previously in Chapter 4. A base model was created per time routine to increase diversity at a data level: Morning, Afternoon and Evening models were generated due to some activities exclusively occurring within specific routines. Furthermore, a Mixed model was generated to consider and represent activities that transpire arbitrarily throughout a typical daily routine. Each NN base model was constructed in Matlab with two hidden layers as [128] states the implementation of a simplistic NN architecture with 2/3 layers can be most effective for HAR tasks. In determining the number of hidden neurons required per base model, a grid search method was utilised. A “rule of thumb” in selecting hidden neurons is that the selection for the first layer should be half of the size of the model inputs, and the selection for the second layer should be halved again, according to [195]. For example, with 31 inputs these values would be around 16 and 8 hidden neurons for layers 1 and 2, respectively. Consequently, the ad-hoc grid search values for layer 1 were between 10 and 20 neurons, and layer 2 were between 5 and 15 neurons, per base model. Figure 6.1 presents the 4 base models where n indicates the number of classes per model. M , A , and E represent the Morning, Afternoon and Evening models, respectively, and finally, MI represents the Mixed model.

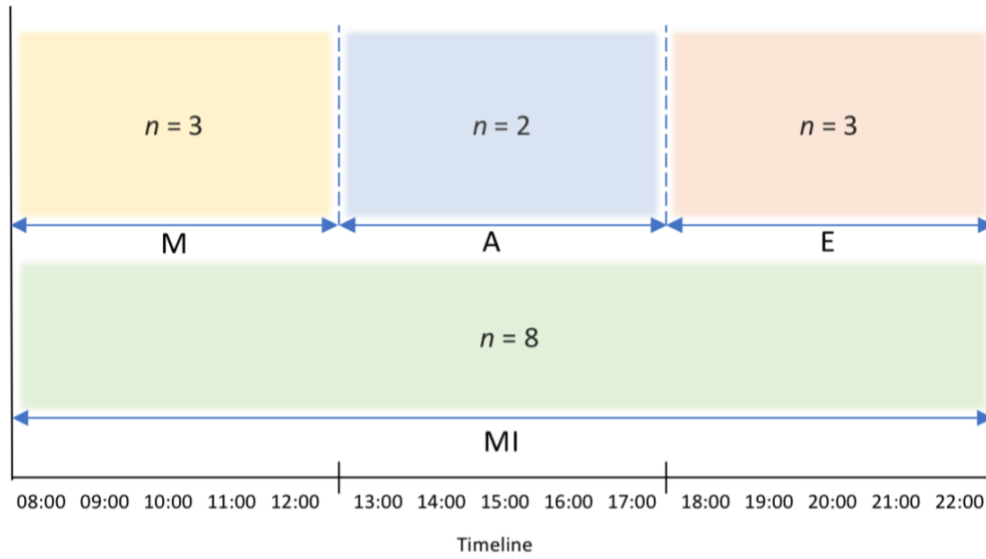


Figure 6.1. Four base classifiers presented per time routine, where n indicates the number of classes per model. M , A , and E represent the Morning, Afternoon, and Evening models, respectively, and finally MI represents the Mixed model.

The morning base model contained two main activity classes, namely Act24 wake up and ActN3 breakfast, as these activities occur in a typical morning routine. ActN4 lunch was the only main class within the afternoon base model as lunch usually occurs in the afternoon. The evening base model contained two main classes, namely Act23 go to bed and ActN5 dinner, as these activities habitually occur during an evening routine. Finally, the mixed base model contained seven main activity classes that do not regularly occur within a specific time routine. For example, Act15 put waste in the bin and Act22 dressing are activities commonly performed at any time throughout a typical day. The activity class outputs per model are presented in Table 6.1.

Table 6.1. Activity class outputs per model.

#output	Model ID	Name	Activity Classes
$m_1 = 3$	M_1	Morning	$C^1 = [\text{Act24}, \text{ActN3}] \leftarrow 2 \text{ classes}$ $\widetilde{C}^1 = [\text{ActN4}, \text{Act23}, \text{ActN5}, \text{Act01}, \text{Act15}, \text{Act17}, \text{Act18}, \text{Act22}, \text{ActN1}, \text{ActN2}] \leftarrow 1 \text{ class}$
$m_2 = 2$	M_2	Afternoon	$C^2 = [\text{ActN4}] \leftarrow 1 \text{ class}$ $\widetilde{C}^2 = [\text{Act24}, \text{ActN3}, \text{Act23}, \text{ActN5}, \text{Act01}, \text{Act15}, \text{Act17}, \text{Act18}, \text{Act22}, \text{ActN1}, \text{ActN2}] \leftarrow 1 \text{ class}$
$m_3 = 3$	M_3	Evening	$C^3 = [\text{Act23}, \text{ActN5}] \leftarrow 2 \text{ classes}$ $\widetilde{C}^3 = [\text{Act24}, \text{ActN3}, \text{ActN4}, \text{Act01}, \text{Act15}, \text{Act17}, \text{Act18}, \text{Act22}, \text{ActN1}, \text{ActN2}] \leftarrow 1 \text{ class}$
$m_4 = 8$	M_4	Mixed	$C^4 = [\text{Act01}, \text{Act15}, \text{Act17}, \text{Act18}, \text{Act22}, \text{ActN1}, \text{ActN2}] \leftarrow 7 \text{ classes}$ $\widetilde{C}^4 = [\text{Act24}, \text{ActN3}, \text{ActN4}, \text{Act23}, \text{ActN5}] \leftarrow 1 \text{ class}$

Each NN base classifier needed to be trained with the inclusion of an additional class, namely the complement, due to each model containing non-overlapping classes. Thus, the complement class, per model, consisted of representative activity samples from each of the main classes contained within the remaining base models. The aim of the complement class was to support each model in identifying whether or not new, unseen activity instances belonged to that particular model. Thus, when an unseen input of an activity class is presented to the considered base model, that exists within its complement, the model should have recognised that the activity does not exist as a main class within that particular model and should, consequently, exclude itself from the decision process. For example, consider the morning model, M_1 , was presented with an activity instance contained in the \widetilde{C}^1 class, such as ActN4 lunch, as presented in Table 6.1. This model should have ideally recognised that ActN4 lunch belonged to the complement class and should therefore have eliminated itself from the decision-making process.

Definitions of the models are described as follows:

Input data X :

$$X = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N]^R \in B^{N \times d},$$

where N is the number of instances, d is the number of features, $d=31$.

$$\vec{x}_i = [x_i^1, x_i^2, \dots, x_i^d] \text{ where } x_i^d \in [0,1].$$

Output:

$$Y = [y_1, y_2, \dots, y_N]^R \in [1, \dots, 12].$$

Base Models:

Base models M_1 , M_2 , M_3 , and M_4 represent the Morning, Afternoon, Evening, and Mixed base models, respectively, in the proposed ensemble approach.

Given the instance \vec{x}_i base model output M_j is given by

$$f_i^j = f^j(\varphi^j(x_i)),$$

where index $j = [1, \dots, 4]$; $\varphi^j(x_i)$ is the input to the activation function of base model M_j and f^j is the output of each base model M_j .

For simplicity, the output can be represented as $f_i^j = [p_1^j, \dots, p_{m_j}^j]$, where m_j represents the number of outputs from base model M_j .

Predicted class $\hat{k}_i^j \in [1, \dots, 12]$ from base model M_j is the class represented by the output with maximum p values $p_i^{j,1} = \max [p_1^j, \dots, p_{m_j}^j]$. The second largest value in the output vector is notated as $p_i^{j,2}$.

Base Model Compositions:

Universal set C represents the set of all classes of activities; C^j represents activity classes represented by the time domain of each base model M_j .

\widetilde{C}^j is the complement class for base model M_j and it combines the activity classes not in the C^j denoted as

$$\{k \in C : k \notin C^j\}$$

Example: Morning Base Model M_1 contains activities from classes

$$C^1 = [Act24, ActN3]$$

$$\widetilde{C}^1 = [\{ActN4, Act23, ActN5, Act01, Act15, Act17, Act18, Act22, ActN1, ActN2\}]$$

There are $m_j = 3$ number of classes, where all but one class, the complement, are in C^1 .

A comprehensive framework pertaining to the implemented homogeneous ensemble method is presented in Figure 6.2, within which each of the conflict resolution techniques were compared. Each NN base classifier was presented with an input feature vector consisting of 31 features. These features comprised data produced by 30 environmental binary sensors, and an additional time routine feature. Each of the base classifiers produced output predictions obtained from the estimated likelihood of each activity class, which were subsequently combined through the support function fusion method [138] during the ensemble integration phase.

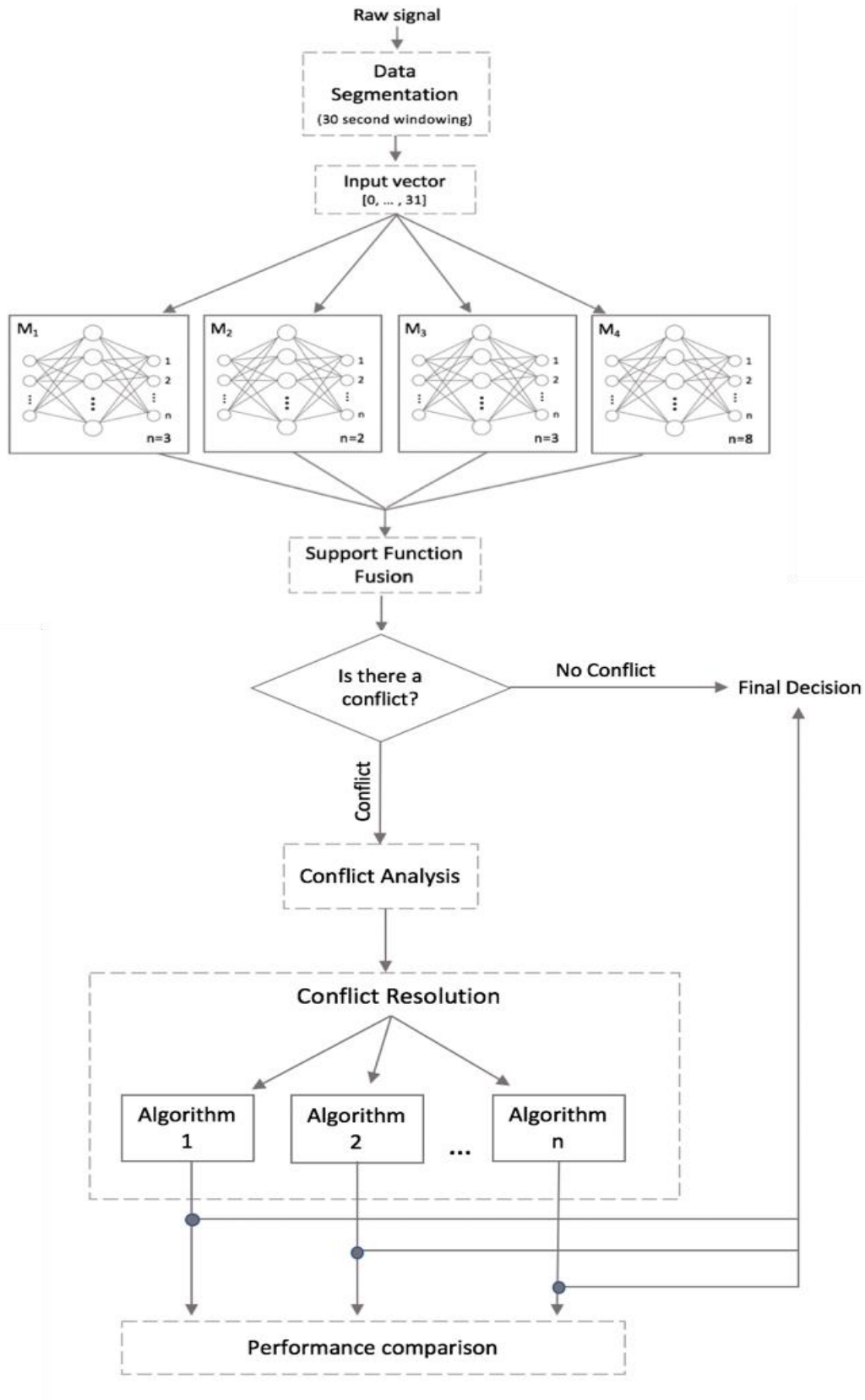


Figure 6.2. Framework for the homogeneous ensemble approach. M_1 , M_2 and M_3 represent the Morning, Afternoon and Evening models, respectively, and M_4 represents the Mixed model.

As mentioned, each NN base classifier was trained with the inclusion of a complement class. To analyse the effects on model conflicts of various data distributions that construct the complement classes per model, two approaches towards generating these classes were explored. Section 6.3.1.1 explains the generation of the complement class data at a model level, where activity instances were distributed evenly between the remaining models, and Section 6.3.1.2 explains the generation of the complement class data at a class level, where activity instances were distributed evenly between the remaining classes.

6.3.1.1 Complement Class Generation at a Model Level

Distributing instances at a model level involved balancing the complement class data equally between the remaining models. The first step in the process was to calculate how many instances this class should contain, in total. Per base model, this was calculated as the average number of main class instances. This number was then divided by the number of remaining models to achieve an equal distribution of activity instances per time routine. Following this, the class distributions were calculated by dividing the number of instances per model by the number of main classes within each model. Table 6.2 presents the distribution of instances at a model level.

Table 6.2. Model level-distribution of instances for complement class compositions, where M , A , E , and MI within the class distributions indicate classes belonging to the Morning, Afternoon, Evening, and Mixed models, respectively.

Complement	Model Distribution (No. of Instances)	Class Distribution (No. of Instances)	
complement class \widetilde{C}^1 of M_1	Afternoon (24) Evening (24) Mixed (25)	A: ActN4 (24)	MI: Act17 (03)
		E: Act23 (12)	MI: Act18 (04)
		E: ActN5 (12)	MI: Act22 (04)
		MI: Act01 (03)	MI: ActN1 (04)
		MI: Act15 (03)	MI: ActN2 (04)
complement class \widetilde{C}^2 of M_2	Morning (62) Evening (62) Mixed (62)	M: Act24 (31)	MI: Act15 (09)
		M: ActN3 (31)	MI: Act17 (09)
		E: Act23 (31)	MI: Act01 (09)
		E: ActN5 (31)	MI: Act22 (09)
		MI: Act18 (08)	MI: ActN1 (09)
complement class \widetilde{C}^3 of M_3	Morning (27) Afternoon (27) Mixed (27)	MI: ActN2 (09)	MI: ActN2 (09)
		M: Act24 (13)	MI: Act15 (04)
		M: ActN3 (14)	MI: Act17 (04)
		A: ActN4 (27)	MI: Act22 (04)
		MI: Act18 (03)	MI: ActN1 (04)
complement class \widetilde{C}^4 of M_4	Morning (24) Afternoon (24) Evening (25)	MI: Act01 (04)	MI: ActN2 (04)
		M: Act24 (12)	E: Act23 (12)
		M: ActN3 (12)	E: ActN5 (12)
		A: ActN4 (24)	

6.3.1.2 Complement Class Generation at a Class Level

Distributing instances at a class level involved balancing the complement class data equally between the remaining classes within the models. As with the previous approach, the first step involved calculating the average number of main class instances per model to attain the number of instances for each complement class. Following this, the previously calculated number was divided by the number of remaining classes across the remaining models to achieve an equal distribution of instances per class. Finally, all instances per class were multiplied by 2 to better represent each class. For example, to generate the M_1 complement class, the average number of main class instances was calculated first, resulting in 74. Subsequently, to achieve an equal distribution of instances per class within the complement class, 74 was divided by the 10 remaining classes, resulting in 7.4 instances required per class. Finally, to better represent each class during training, this number was

multiplied by 2, resulting in 14.8 (15) instances per class. Table 6.3 presents the distribution of instances at a class level.

Table 6.3. Class level-distribution of instances for complement class compositions, where M , A , E , and MI within the class distributions indicate classes belonging to the Morning, Afternoon, Evening, and Mixed models, respectively.

Complement	Model Distribution (No. of Instances)	Class Distribution (No. of Instances)	
complement class \widetilde{C}^1 of M_1	Afternoon (15) Evening (30) Mixed (105)	A: ActN4 (15)	MI: Act17 (15)
		E: Act23 (15)	MI: Act18 (15)
		E: ActN5 (15)	MI: Act22 (15)
		MI: Act01 (15)	MI: ActN1 (15)
		MI: Act15 (15)	MI: ActN2 (15)
complement class \widetilde{C}^2 of M_2	Morning (68) Evening (68) Mixed (238)	M: Act24 (34)	MI: Act15 (34)
		M: ActN3 (34)	MI: Act17 (34)
		E: Act23 (34)	MI: Act01 (34)
		E: ActN5 (34)	MI: Act22 (34)
		MI: Act18 (34)	MI: ActN1 (34)
complement class \widetilde{C}^3 of M_3	Morning (32) Afternoon (16) Mixed (112)	MI: ActN2 (34)	MI: ActN2 (34)
		M: Act24 (16)	MI: Act15 (16)
		M: ActN3 (16)	MI: Act17 (16)
		A: ActN4 (16)	MI: Act22 (16)
		MI: Act18 (16)	MI: ActN1 (16)
complement class \widetilde{C}^4 of M_4	Morning (58) Afternoon (29) Evening (58)	MI: Act01 (16)	MI: ActN2 (16)
		M: Act24 (29)	E: Act23 (29)
		M: ActN3 (29)	E: ActN5 (29)
		A: ActN4 (29)	

6.3.2 Model Conflict Resolution

As previously mentioned, support function integration [138] was applied to combine the output predictions generated by each NN base classifier during the ensemble integration phase. Subsequently, conflict analysis was performed upon the combined output predictions to ascertain whether a single base model had chosen the final class output, indicating that all models except one had chosen the complement class. If this did not occur, and more than one model had chosen a main class output, a conflict had transpired between these models during the decision-making process, as demonstrated in Algorithm 6.1.

Algorithm 6.1. Process of finding conflicts between models

-
- 1: For Each instance $\vec{x}_i \in B^{1 \times d}$
 - 2: if $\exists_j (\hat{k}_i^j \in C^j) \wedge \exists_{jj} (\hat{k}_i^{jj} \in C^{jj} \wedge j \neq jj)$
 - 3: Then apply conflict resolution approaches in Algorithms 2/3/4/5 as there are
conflicting cases between base models M_j and M_{jj}
-

Several methods to resolve conflicts occurring between base models were explored to ascertain the final output class per activity instance. The first conflict resolution method, presented in Algorithm 6.2, was simply to award the final decision to the model with the highest output prediction. This approach had previously been established as a soft-level combiner [196], as it makes use of the output predictions given by the classifiers as the posterior probabilities of each output class. A limitation of this method, however, was that it provided limited confidence of the output prediction. For example, consider the two largest output values of one base model were 0.52 and 0.48, respectively. If the final class decision had been awarded according to the highest output value in this case, there is less confidence in the quality of classification, which implies a less secure output prediction.

To overcome this, another technique, presented in Algorithm 6.3, was proposed to calculate the difference between the highest and second highest predictions per conflicting model, where subsequently the final decision was given to the model with the highest differential value, as this was deemed the model with the strongest class prediction.

Following this, the impact of a weighting technique was investigated in Algorithm 6.4 on the basis of the number of classes per model, as each base model contained a different number of unique classes. This approach considered the output predictions from each conflicting base classifier and the number of classes the base models were trained on, specifically, the output predictions from each base model were multiplied by the number of classes within those base models. For example, if a conflict occurred between model M_2 and model M_4 , which contained two and eight classes, respectively, the two class problem may be less complex than the eight class problem, and therefore a lower weighting was specified for M_2 .

Finally, the potential of another weighted method in Algorithm 6.5 was explored, which built upon the previous approach. Weightings were implemented

on the basis of the number of classes, as well as the training performance per model, specifically, the output predictions from each conflicting base classifier were multiplied by the number of classes in that model and the training performance achieved. According to [197], a base classifier that outperforms other base classifiers in an ensemble approach should be given a higher confidence when deciding upon the final output prediction, as the training performance measure is indicative of the classifier's effectiveness in predicting the correct output class. The training performance measure in Algorithm 6.5 was the classification accuracy obtained by each conflicting model when exposed to the training set.

Repeated notations:

The largest value in the output vector is notated as $p_i^{j,1}$.

The second largest value in the output vector is notated as $p_i^{j,2}$.

Algorithm 6.2. Conflict resolution approach 1, where the model with highest prediction is awarded the output decision

Input: \vec{x}_i , base models M_r, M_s

Output: class y_i

1:	$if\ p_i^{r,1} > p_i^{s,1}$
2:	Then $y_i = \hat{k}_i^r$
3:	Else $y_i = \hat{k}_i^s$

Algorithm 6.3. Conflict resolution approach 2, where the model with the highest differential value is awarded the output decision

Input: \vec{x}_i , base models M_r, M_s

Output: class y_i

1:	$if\ (p_i^{r,1} - p_i^{r,2}) > (p_i^{s,1} - p_i^{s,2})$
2:	Then $y_i = \hat{k}_i^r$
3:	Else $y_i = \hat{k}_i^s$

Algorithm 6.4. Conflict resolution approach 3, where the model with the highest value through multiplying the output prediction by the number of classes is awarded the output decision

Input: \vec{x}_i , base models M_r, M_s

Output: class y_i

1:	$if\ p_i^{r,1} \times m_r > p_i^{s,1} \times m_s$
2:	Then $y_i = \hat{k}_i^r$
3:	Else $y_i = \hat{k}_i^s$

Algorithm 6.5. Conflict resolution approach 4, where the model with the highest value through multiplying the output prediction by the number of classes and the training performance is awarded the output decision

Input: \vec{x}_i , base models M_r, M_s

Output: class y_i

1:	Acc_{train}^r represents training performance for base model M_r
2:	Acc_{train}^s represents training performance for base model M_s
3:	$if\ p_i^{r,1} \times m_r \times Acc_{train}^r > p_i^{s,1} \times m_s \times Acc_{train}^s$
4:	Then $y_i = \hat{k}_i^r$
5:	Else $y_i = \hat{k}_i^s$

6.4 Results and Discussion

The obtained results demonstrated that the class level distribution technique, described in Section 6.3.1.2, greatly reduced the number of conflicts that occurred between the various base models, in comparison to the model level distribution technique, as presented in Table 6.4. This was due to improved representations of activities within the complement classes per model during the training phase of the base classifiers. For example, with the class level distribution technique, activity instances were distributed evenly between classes, therefore evenly representing each activity within the complement class. Contrarily, the model level distribution technique involved balancing the complement class data equally between the remaining models, which meant the class distributions within these models were imbalanced. For example, with the model level distribution technique, the \widetilde{C}^1 complement class contained 24 instances of ActN4 lunch and only 03 instances of Act17 brush teeth, whereas with the class level distribution technique, the \widetilde{C}^1 complement class contained 15 instances each of ActN4 lunch and Act17 brush teeth. Consequently, with the implementation of the latter distribution technique, the base classifiers were stronger at deciding when an unseen instance belonged to their complement class, eliminating themselves from the decision-making process, and therefore reducing the number of conflicts that occurred.

Table 6.4. Number of conflicts occurring, per fold, through each data distribution of the complement class.

	No. of Conflicts Per Fold										Avg.
	1	2	3	4	5	6	7	8	9	10	
Complement Class – Model Level Approach	76	57	69	52	49	35	60	45	62	56	56.1
Complement Class – Class Level Approach	21	37	11	13	13	42	29	39	11	17	23.3

Figure 6.3 presents the classification performances achieved through each of the data distribution techniques, which were analysed before and after the application of conflict resolution methods. Considering the complement class generation at a model level, the preliminary classification accuracy of 60.28% was much less than that of the complement class generation at a class level, which achieved a preliminary accuracy of 72.12%. This was due to less model conflicts occurring in the latter approach, which demonstrated that the base models were stronger during the decision-making process. As for the final accuracies produced after conflict resolution techniques had been applied, the class level approach outperformed the model level approach in all four cases. Finally, overall, the best HAR performance of 80.39% was achieved using complement data generated at a class level in conjunction with the conflict resolution approach presented in Algorithm 6.3 as described in Section 6.3.2.

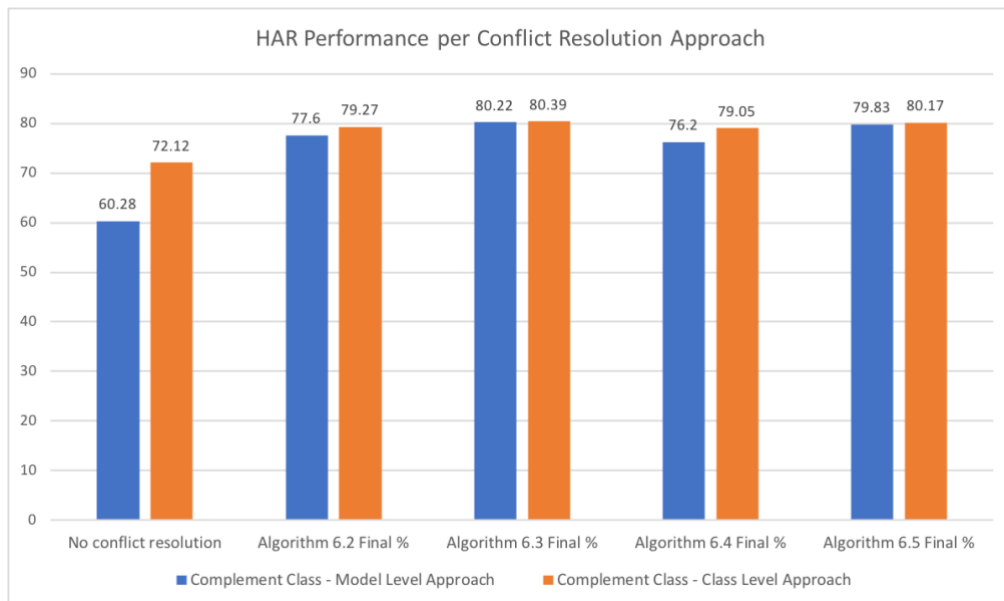


Figure 6.3. Human Activity Recognition (HAR) performance per conflict resolution approach.

Table 6.5 presents an analysis of incorrectly classified instances with regards to the first data distribution approach where complement class data was generated at a model level, as discussed previously in Section 6.3.1.1, whereas Table 6.6 presents an analysis of incorrectly classified instances with regards to the second data distribution approach, where complement class data was generated at a class level, as discussed previously in Section 6.3.1.2. The “incorrect” instances reported describe those that were incorrectly classified by the target model, for example, there may not have been any conflicting models, yet the incorrect class was chosen by the base classifier. The number of incorrectly classified instances were important to consider when analyzing the effectiveness of each conflict resolution approach, as these cases would have been permanently incorrect, regardless of the application of conflict resolution techniques.

The “right but incorrect” cases were those that were correctly classified by the target base model, though were not chosen during the final decision-making process after applying the conflict resolution approaches. These cases were considered when evaluating the most effective approach of the four explored, as they could have resulted in a correct classification, given the application of an effective conflict resolution technique.

Table 6.5. Ensemble approach 1 - Analysis of incorrect instances, where A.6.2, A.6.3, A.6.4 and A.6.5 represent the Algorithm number

		Fold										Avg.
		1	2	3	4	5	6	7	8	9	10	
A.6.2	Incorrect	22	22	21	29	29	20	30	22	20	22	23.7
	Right but Incorrect	17	18	21	12	17	16	9	14	20	20	16.4
A.6.3	Incorrect	23	22	21	29	29	22	29	22	20	24	24.1
	Right but Incorrect	10	14	10	9	12	12	9	12	14	11	11.3
A.6.4	Incorrect	22	23	21	29	29	22	29	22	20	22	23.9
	Right but Incorrect	31	22	13	23	11	15	23	18	10	21	18.7
A.6.5	Incorrect	22	22	21	29	29	22	29	22	20	22	23.8
	Right but Incorrect	14	10	13	7	13	15	9	17	14	11	12.3

Table 6.6. Ensemble approach 2 – Analysis of incorrect instances, where A.6.2, A.6.3, A.6.4 and A.6.5 represent the Algorithm number

		Fold										Avg.
		1	2	3	4	5	6	7	8	9	10	
A.6.2	Incorrect	33	26	35	33	25	32	27	28	40	26	30.5
	Right but Incorrect	6	9	2	6	4	11	8	10	2	8	6.6
A.6.3	Incorrect	33	26	35	33	25	31	27	28	40	26	30.4
	Right but Incorrect	5	7	3	2	6	7	6	5	0	6	4.7
A.6.4	Incorrect	33	26	35	33	25	31	27	28	40	25	30.3
	Right but Incorrect	8	21	4	3	6	6	11	7	1	5	7.2
A.6.5	Incorrect	33	26	34	33	25	31	27	28	40	25	30.2
	Right but Incorrect	8	8	5	2	6	5	6	7	1	5	5.3

The conflict resolution approach presented in Algorithm 6.3 was the most effective when applied to both data distributions, as there were the lowest number of “right but incorrect” instances (on average 11.3 and 4.7, respectively), closely followed by the approach in Algorithm 6.5. The lower the number of “right but incorrect” cases helped to determine which conflict resolution approach was most effective in deciding upon which base model should be awarded the final class decision. For example, consider the conflict resolution technique in Algorithm 6.3 with ensemble approach 2, as presented in Table 6.6. There were 23.3 conflicts occurring on average (refer to Table 6.4). Upon analysis of the incorrectly classified instances, 30.4, on average, were incorrectly classified, whereas 4.7, on average, could have been correctly classified, though an incorrect base model won the final decision after applying conflict resolution. Finally, this meant that as a result of applying Algorithm 6.3, an average of 18.6 conflicting cases were correctly resolved, improving the final HAR performance.

As presented in Figure 6.3, the best HAR performance of 80.39% was achieved using complement data generated at a class level in conjunction with the conflict resolution approach presented in Algorithm 6.3. This optimally performing ensemble method was subsequently benchmarked against the suite of classifiers

previously introduced in Chapter 5, namely kNN, SVM, NN and LR models. Figure 6.4 presents the performance of the proposed ensemble approach in comparison to the defined suite of classifiers. The kNN model had achieved an accuracy of 70.95%, whereas the SVM model achieved 76.54%, thus demonstrating that the proposed ensemble approach outperformed 2 of the 3 considered non-parametric classifiers. The NN model had very slightly outperformed the ensemble method, comparatively demonstrating a performance increase of 0.06%. Finally, the LR model had slightly outperformed the proposed method by 0.62%.

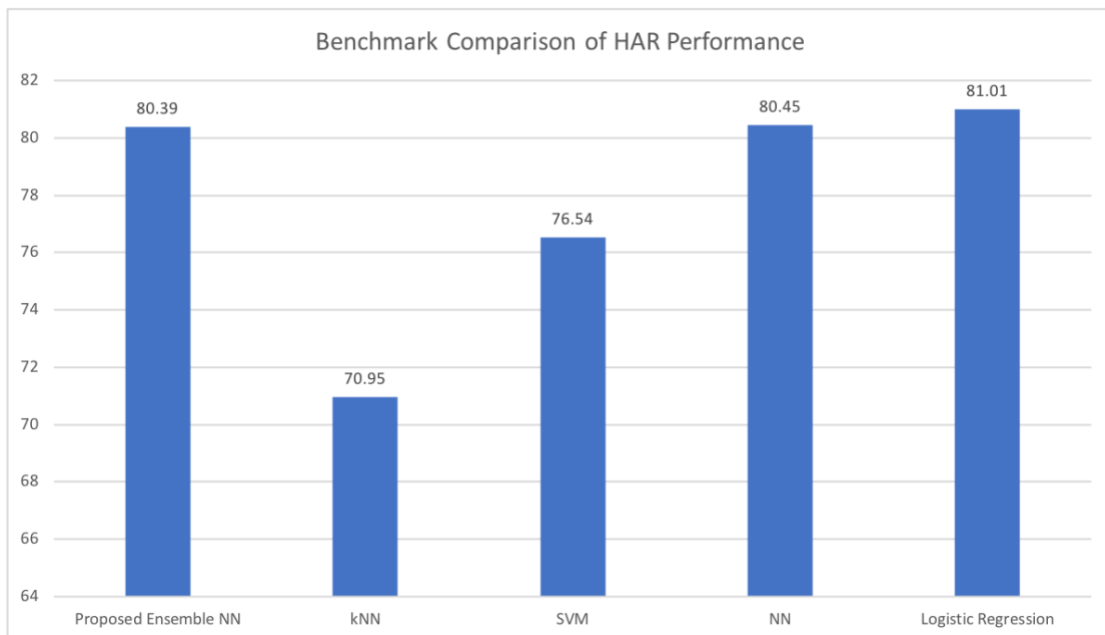


Figure 6.4. HAR performance of the proposed ensemble NN approach compared to kNN, SVM, NN, and Logistic Regression classifiers, in terms of accuracy (%).

6.5 Conclusion

As stated, the widely acknowledged desire to explore ensemble methods derives from their ability to enhance classification performance and their potential to improve generalisation capabilities through diminishing the limitations of individual classifiers. Ensemble generation and integration are two primary aspects to consider whilst exploring ensemble methods, thus experimentation within this Chapter involved investigations into both aspects. A novel ensemble approach was proposed to recognize ADLs within a smart environment setting, with four NN base classifiers created to represent time routines throughout a typical day, including Morning,

Afternoon, and Evening routines. Additionally, a Mixed routine was included to represent activities occurring throughout the day. Particular emphasis was made upon analyzing the effects of various data distributions that generate the complement class per base model during the ensemble generation phase, and exploring various approaches to resolving conflicts that occur between base models during the ensemble integration phase. Thus, two data distribution techniques were explored, including the generation of complement class data at a class level, and at a model level. Furthermore, four conflict resolution techniques were investigated, including awarding the final output decision to the model with the highest output prediction, the highest differential value, or weighting techniques involving the number of classes per model, and training performance achieved².

Through analysis of experimental results, it was observed that distributing data at a class level greatly reduced the number of conflicts that occurred between the base models, leading to an increased preliminary performance before the application of conflict resolution techniques. It was also found that the best method of resolving conflicts, in comparison to other approaches explored, was to award the final decision to the model with the highest differential value between the highest and second highest predictions per conflicting model. The proposed HAR classification model, the ensemble NN method, was evaluated through comparing the achieved HAR performance with three non-parametric benchmark classifiers, namely SVM, NN and kNN models, and an LR model. The ensemble NN method outperformed both the kNN and SVM benchmark classifiers, demonstrating the effectiveness of the proposed ensemble approach, however, the remaining benchmark classifiers very slightly outperformed the proposed ensemble method. Thus, Chapter 7 further explores ensemble classifiers in terms of increasing diversity, as ensemble diversity has been recognised as an essential condition in generating an adequate ensemble [51] and according to [130] the diversity introduced through constructing heterogeneous ensembles appeared promising. Thus, Chapter 7 will explore the potential of heterogeneous ensemble classifiers to enhance HAR classification performance.

² The results in this Chapter were published in [206]

Chapter 7

Heterogeneous Ensembles for Human Activity Recognition

7.1 Overview

Research explorations conducted within this Chapter extend upon the previously implemented ensemble works conducted in Chapter 6. Previously, a homogeneous NN ensemble was implemented due to the perceived benefits of ensemble learning, which conclusively demonstrated their superiority in comparison to two of the benchmarked single classification models, however, the remaining two benchmarked models had very slightly outperformed the proposed homogeneous ensemble. Diversity was achieved at a data level through diversifying the input data to each NN base classifier, however, it has since been recognised that achieving diversity through generating diverse base classifiers within heterogeneous

ensembles, has demonstrated additional benefits in comparison to homogeneous ensemble methods [198], [199]. According to [51], diversity is a critical condition in constructing adequate ensembles, thus, a motivating factor for experimentation conducted within the current Chapter has been to further increase diversity through also varying the choice of base classifiers in an endeavour to further enhance HAR performance.

Within this Chapter, diversity has been achieved at both a data and classifier level. The models previously created through decomposing the HAR activity classes into base models per time routine, specifically, Morning, Afternoon, Evening, and Mixed, were retained within the current work, along with also generating diverse base classifiers. Two different methods of generating heterogeneous ensembles were explored. The first method involved generating five base classifiers, per base model, within which the best performing base classifier was chosen per time routine. Following this, the output class derived through each base model were evaluated and combined, and finally, conflict analysis and resolution were performed. The second method involved the implementation of a 2-phase heterogeneous ensemble framework. In Phase 1, a heterogeneous ensemble was generated and combined using a hard-level combination method within the existing base models: Morning, Afternoon, Evening, and Mixed. In Phase 2, the output class from each base model, per time routine, was evaluated and combined. Following this, conflict analysis and resolution techniques were applied to decide upon the final output class across the 4 unique base models.

The remainder of this Chapter is structured as follows: Section 7.2 provides the rationale for the exploration of heterogeneous ensembles, Section 7.3 provides the methodology undertaken to implement the proposed heterogeneous ensembles, and Section 7.4 presents the experimental results obtained. Finally, Section 7.5 concludes this Chapter.

7.2 Rationale for Heterogeneous Ensemble

Heterogeneous ensembles have recently attracted considerable research interest due to their performance superiority in comparison to homogeneous methods [177], as generating diverse base classifiers offer additional benefits through

providing different biases and internal representations [200]. According to [130], less exploration had been made with heterogeneous ensembles in the research community due to difficulties existing in controlling interactions between the various learning processes of diverse base classifiers, yet [130] also stated the diversity introduced through constructing heterogeneous ensembles appeared promising. In [201], a heterogeneous ensemble approach was implemented to recognise various activities within the CASAS smart home testbeds. The ensemble included four base classifiers, which included a Hidden Markov Model (HMM), a NN, an SVM, and Conditional Random Fields (CRF). The results were promising and revealed performance improvements over the use of a single classification model. Further to this, [200] implemented an ensemble classification approach to activity recognition using several heterogeneous base classifiers. The five common base classifiers included an SVM, DT, kNN, NN, and NB. Results demonstrated that the ensemble approach combined through majority voting performed extremely well in classifying twelve activities. In a study conducted by [198], efficiency comparisons were made between heterogeneous and homogeneous ensembles for the purpose of cancer diagnosis. The implemented homogeneous ensembles included those generated through combining multiple NNs, Random Forest, SVM and Genetic Algorithms, whereas the heterogeneous ensemble involved combining all four of the aforementioned base classifiers. Experimental results demonstrated the superior classification performance obtained by the heterogeneous ensemble in comparison to all evaluated homogeneous methods, which outperformed each homogeneous method in terms of classification accuracy by at least 2% during experiments pertaining to breast cancer and melanoma diagnosis. As for respiratory system cancer diagnosis, the heterogeneous ensemble and the homogeneous NN ensemble performed equally best in comparison to all other ensemble compositions explored. Additionally, in [199], adaptive heterogeneous ensemble methods based upon DT, NB and kNN classifiers were explored, and performance comparisons were made to the homogeneous methods. Within the proposed adaptive framework, both the size of the generated heterogeneous ensembles (determined by the number of base models included) and the classifier combinations were optimized during training. Experimental results of this study demonstrated the effectiveness of heterogeneous ensembles in comparison to their homogeneous counterparts, whilst suggesting their superiority was due to the complementary nature of combining diverse classifiers,

particularly when evaluated on multiclass classification tasks consisting of more than 3 class labels.

More recently, introducing diversity has been recognised as an essential condition in generating an adequate ensemble [51]. Nevertheless, according to [51] no formal definition pertaining to diversity exists, and no clear consensus exists as to how diversity can be measured. It has, however, been recognised that generating dissimilar decisions through achieving diversity, may occur through either altering the training data and/or classifiers [51]. Given that diversity was previously achieved through altering only the data in Chapter 6, an opportunity has emerged to increase diversity further by also varying the choice of base classifiers.

7.3 Proposed Heterogeneous Ensemble Methods

As presented in Chapter 6, diversity was achieved through organising the training data uniquely amongst the base models, which were constructed based upon time routines: Morning (M_1), Afternoon (M_2), Evening (M_3), and Mixed (M_4). As each model contained unique activity classes, each was trained with an additional complement class comprising data samples used to train the remaining models. Thus, the proposed heterogeneous ensemble methods extend the level of diversity introduced by generating diverse base classifiers. Previously, two methods of generating the complement class per model were explored: generating complement data at a class level, and at a model level. Nevertheless, experimental results demonstrated the effectiveness of the class-level distribution method, which involved generating the complement class instances through balancing them equally amongst the remaining classes. Thus, this technique will be implemented with the proposed heterogeneous approaches within this Chapter.

The 5 chosen base classifiers to introduce diversity are well-established classification algorithms [200], namely an SVM, NN, NB, DT, and kNN. This same group of diverse classifiers had been assembled in a recently conducted heterogeneous ensemble study, with promising results achieved through a majority voting combination method [200]. Thus, suggesting the range of base classifiers involved in this work may prove complementary. The base classifiers were trained within Matlab through employing 10-fold cross-validation and performance was measured

in terms of classification accuracy obtained. Each classifier was configured to conform with previous Chapters through employing the same configuration method of selecting the optimally performing model parameters recommended within Matlab. Figure 7.1 presents the heterogeneous ensemble process undertaken per base model, M_1 to M_4 , to generate the diverse base classifiers, particularly, an example of the ensemble process for base model M_1 is described within which 5 base classifiers were generated, specifically M_{11} to M_{15} . This process was repeated to select the base classifiers for M_2 , M_3 , and M_4 .

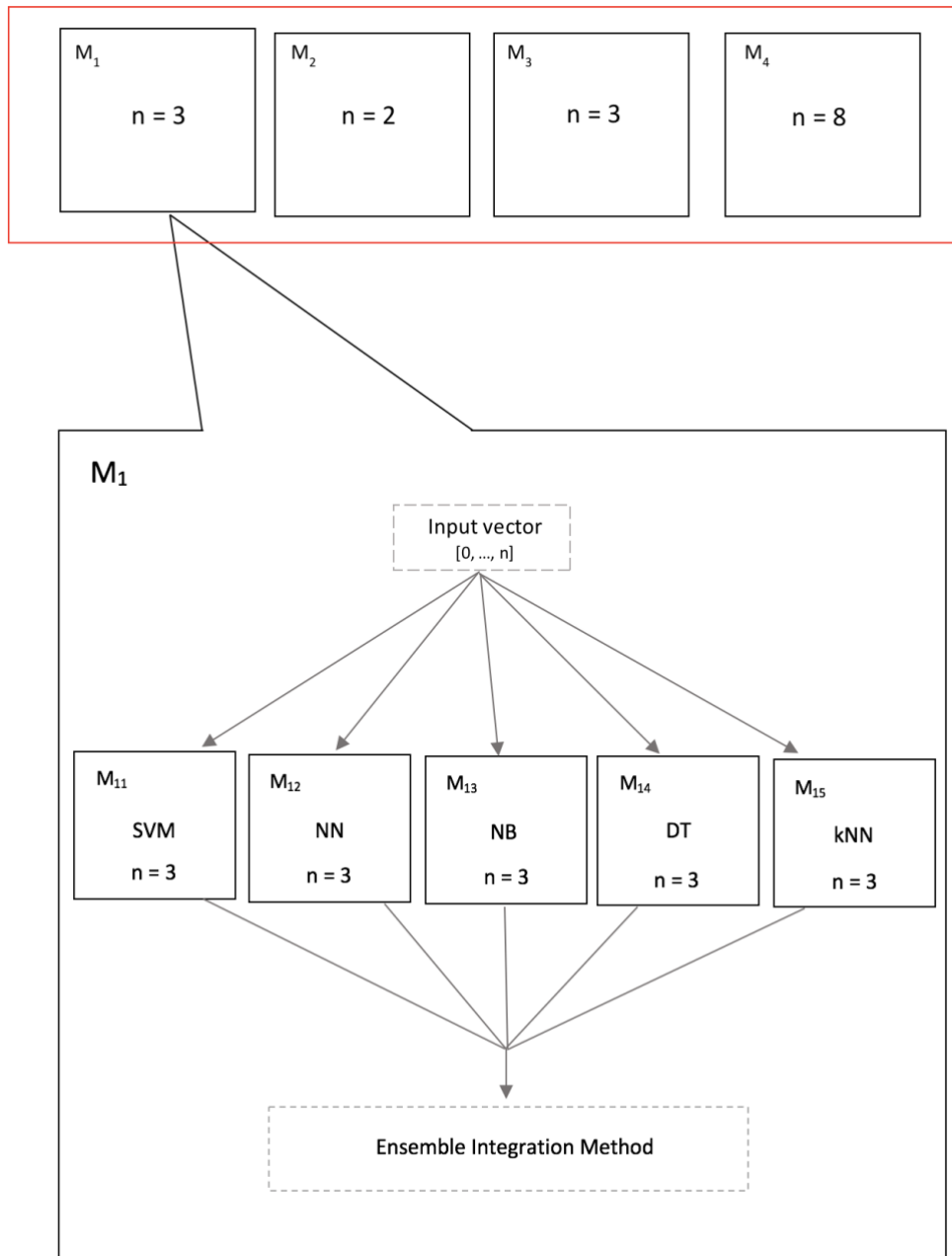


Figure 7.1. The heterogeneous ensemble generation process for M_1 , where n indicates the number of classes per model.

Two diverse methods were explored to produce the output decision vector of the heterogeneous ensemble. Thus, Section 7.3.1 describes the initially explored method of generating a heterogeneous ensemble (Ensemble Method 1). Section 7.3.2 builds upon this and describes the second method considered (Ensemble Method 2).

7.3.1 Ensemble Method 1

The first heterogeneous ensemble method initially involved generating the 5 aforementioned diverse base classifiers, per time routine, to build the base models M_1 to M_4 . The training performances achieved by each classifier were then compared in terms of classification accuracy achieved to ascertain which of the five evaluated base classifiers would be chosen to represent the considered base model, M_1 to M_4 respectively, as the performance achieved during the training phase is indicative of the base classifiers' competence in predicting the accurate class label during testing. For example, during base model composition for M_1 , the best performing classifier was the SVM, thus, the SVM was chosen as the base classifier for the M_1 base model, and the remaining classifiers were disregarded. Figure 7.2 depicts the heterogeneous ensemble implemented through Method 1, which as an example demonstrates how the base classifier for base model M_1 was chosen. This process was repeated to select the base classifier for models M_2 , M_3 , and M_4 .

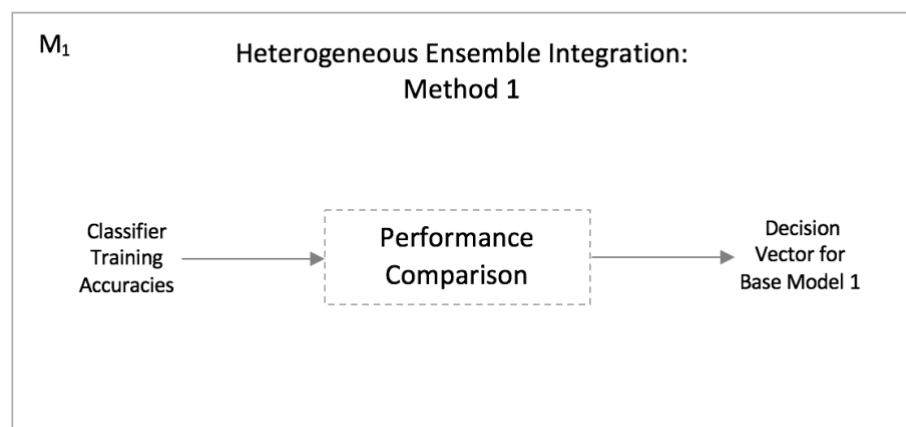


Figure 7.2. Heterogeneous Ensemble Method 1

Table 7.1 presents the training performances achieved by each evaluated classifier. As stated, the classifier achieving the best performance was chosen to represent each base model. Consequently, an SVM was chosen to represent the M_1 model, a NB was chosen to represent the M_2 model, an NN was chosen to represent the M_3 model, and finally, a DT was chosen to represent the M_4 model. Models M_1 , M_2 and M_3 all achieved substantially high training accuracies in comparison to model M_4 , which achieved a considerably lower performance. This may be due to the number of classes involved per base model. For example, M_2 contains 2 classes whereas M_4 contains 8 classes, thus the 2-class problem is deemed substantially less intricate than the 8-class problem.

Table 7.1. Training performances achieved by each base classifier

Base Model	Base Classifier	Train Performance (%)
M_1	SVM (M_{11})	98.32
	kNN (M_{12})	97.64
	NN (M_{13})	96.93
	DT (M_{14})	97.60
	NB (M_{15})	97.98
M_2	SVM (M_{21})	95.89
	kNN (M_{22})	94.82
	NN (M_{23})	95.89
	DT (M_{24})	95.18
	NB (M_{25})	96.43
M_3	SVM (M_{31})	94.14
	kNN (M_{32})	93.83
	NN (M_{33})	95.07
	DT (M_{34})	94.44
	NB (M_{35})	93.21
M_4	SVM (M_{41})	74.36
	kNN (M_{42})	71.17
	NN (M_{43})	74.96
	DT (M_{44})	75.27
	NB (M_{45})	73.60

The predicted class outputs of the unseen test data per base model were subsequently combined and analysed to ascertain whether conflicts had occurred between each model, indicating that more than one base model had chosen its unique main class output. In the previously implemented homogeneous ensemble approach, four conflict resolution algorithms were implemented and compared based upon the output prediction value of the estimated class likelihood. The conflict resolution approaches in the current heterogeneous work vary from those previously explored as only the class output per classifier is considered as a hard-level combiner, rather than the prediction value of the output class as a soft-level combiner. This was to avoid difficulties in controlling the interactions between each base classifier, as this is recognised as a challenge in integrating heterogeneous base models, according to [130]. Thus, within Ensemble Method 1, 3 conflict resolution techniques were explored based upon adaptations of the previous algorithms.

Conflict resolution approach 1, presented in Algorithm 7.1.1, involved awarding the final output decision to the base model achieving the best classification performance during training, as according to [197], higher confidence should be given to the base model outperforming others during training whilst determining the final output decision. The second conflict resolution approach, presented in Algorithm 7.1.2, involved awarding the final output decision to the base model containing the largest number of activity classes, as each model comprised of various unique classes, and the more classes involved indicate a more complex problem was considered. For example, considering a conflict emerging between models M_3 and M_4 , containing 3 and 8 classes, respectively, the final output decision would be awarded to M_4 , as previously identified, the 3-class problem may be less intricate than the 8-class problem. The final conflict resolution approach, presented in Algorithm 7.1.3, builds upon both Algorithms 7.1.1 and 7.1.2 by awarding the final output decision based upon both criteria, specifically the training performance multiplied by the number of classes per model.

Algorithm 7.1.1 Conflict resolution approach 1, Ensemble Method 1.

Input: \vec{x}_i , base models M_r, M_s

Output: class y_i

- | | |
|----|-------------------------------------|
| 1: | $if\ Acc_{train}^r > Acc_{train}^s$ |
| 2: | Then $y_i = \hat{k}_i^r$ |
| 3: | Else $y_i = \hat{k}_i^s$ |
-

Algorithm 7.1.2 Conflict resolution approach 2, Ensemble Method 1.**Input:** \vec{x}_i , base models M_r, M_s **Output:** class y_i

1:	$if\ m_r > m_s$
2:	Then $y_i = \hat{k}_i^r$
3:	Else $y_i = \hat{k}_i^s$

Algorithm 7.1.3 Conflict resolution approach 3, Ensemble Method 1.**Input:** \vec{x}_i , base models M_r, M_s **Output:** class y_i

1:	$if\ m_r \times Acc_{train}^r > m_s \times Acc_{train}^s$
2:	Then $y_i = \hat{k}_i^r$
3:	Else $y_i = \hat{k}_i^s$

7.3.2 Ensemble Method 2

This ensemble method extends the previous approach through the inclusion of well-established hard-level combination techniques to determine the output predictions per base model. Previously in Method 1, only the base classifier achieving the highest classification performance was chosen to represent each base model, per time routine. Nevertheless, a limitation to this approach was identified in that through disregarding the output class predictions of the remaining classifiers, potentially valuable information provided by those could be lost. Consequently, this method involves the implementation of hard-level combination techniques to combine the predictions of each of the 5 base classifiers, thus strengthening the initial output decision per base model. Given that this method involves two ensemble integration stages, a two-phase framework has been proposed.

Within Phase 1 of the proposed heterogeneous ensemble, 5 diverse base classifiers were generated per time routine and subsequently combined using hard-level combination methods. Figure 7.3 depicts heterogeneous ensemble method 2, in which voting schemes are used to generate a decision vector for M_1 . This process was repeated for models M_2 , M_3 and M_4 .

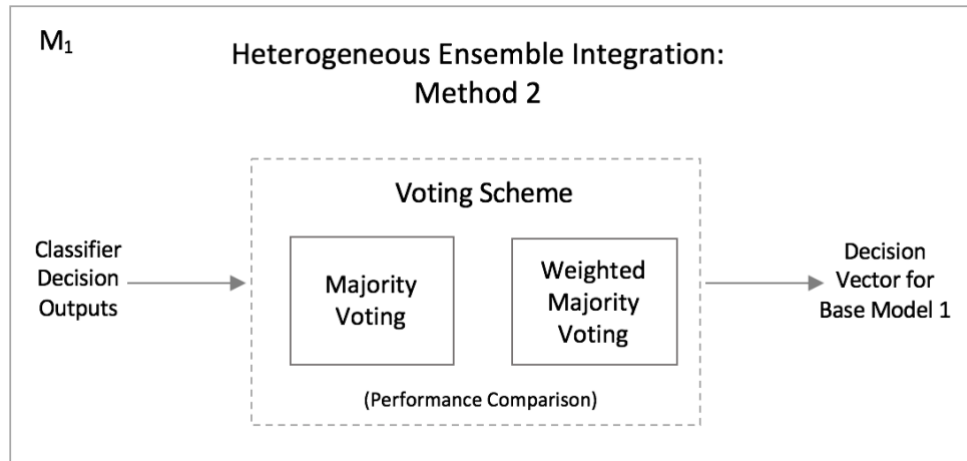


Figure 7.3. Heterogeneous Ensemble Method 2

In the previously implemented homogeneous study within Chapter 6, ensemble integration using a soft-level fusion combination technique was implemented (support function fusion) based upon the output prediction values of the base classifiers. In the proposed heterogeneous approach, hard-level combination methods were implemented to diminish any difficulties existing in controlling interactions between the diverse learning processes of each base classifier. For example, the range of output values differ amongst the diverse base classifiers, as the NN output prediction values vary between 0 and 1 representing the likelihood of each class output, whereas the SVM outputs are represented as posterior probabilities of the class likelihood ranging between -1 and 1. Soft-level combiners are those that consider the likelihood of the output class label as the predictive score, and utilise methods such as sum, max, and min [196]. Whereas, hard-level combiners utilise the output class labels of each base model, for example, through majority voting [196]. The well-established hard-level combination methods to be implemented and compared in Phase 1 are Majority Voting [49] and Weighted Majority Voting [49], [196].

Majority Voting is a commonly implemented ensemble integration method due to its simplicity and effectiveness. This method was designed to fuse the output predictions derived through each base model into a single output whilst taking all base models into consideration [49]. The output class per data instance is compared across all base models, namely the SVM, NN, DT, NB and kNN, thus the final output

class is chosen as that with the largest number of votes. According to [202], majority voting is beneficial in reducing bias and variance, thus improving classification performance. An example of the majority voting implementation is provided in Figure 7.4, within which the class outputs produced by each base classifier were passed into the majority voting module to ascertain the final majority voted output class.

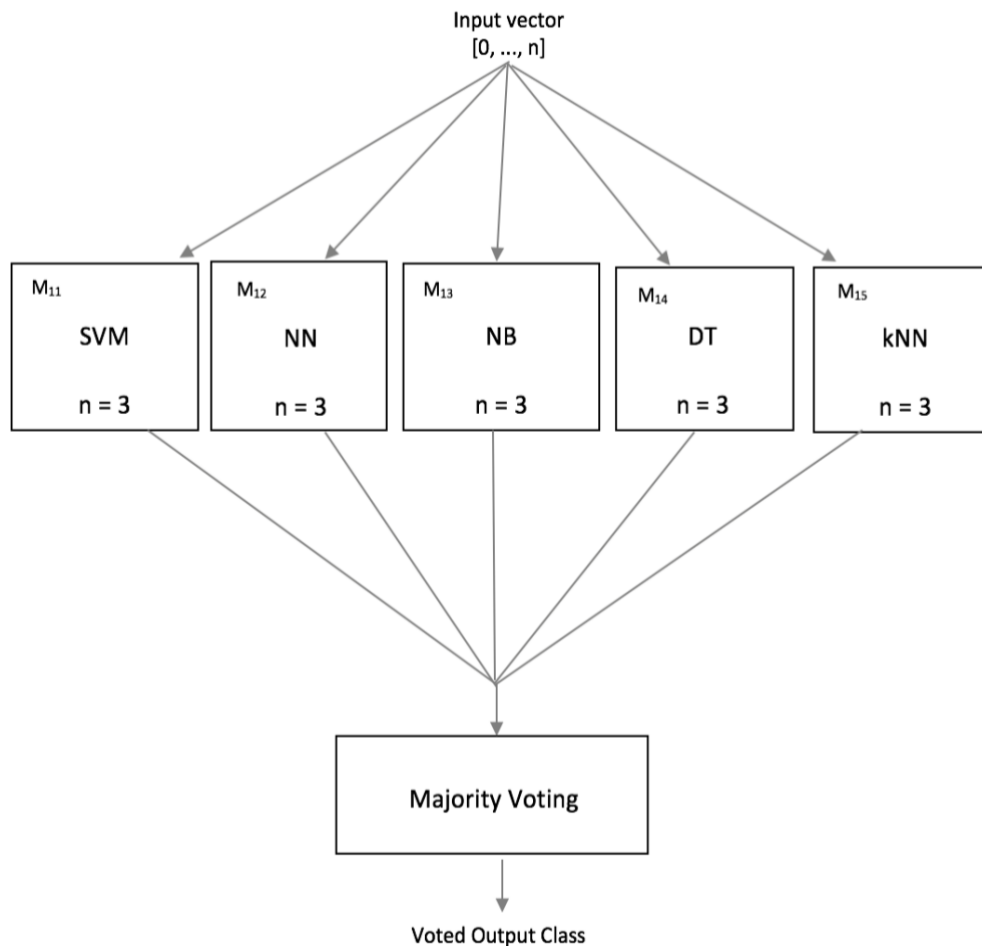


Figure 7.4. Majority Voting implementation where the class outputs from each base classifier are passed into the majority voting module to ascertain the final voted output class

Weighted Majority Voting is an extension of the previously described voting method in which additional weighting is applied to the output predictions of each base classifier. Weightings are commonly assigned based upon the strengths of each classifier within the ensemble, for example the classification performance achieved by each model during training is indicative of their predictive power, thus further supporting the decision upon the final output class [49]. Due to this, the weightings applied within this ensemble were based upon the training performances in terms of

accuracy achieved of each classifier. According to [202], if each base classifier has been assigned very similar weight values, their votes may result in equal value, however, if varied weights are applied according to each base classifier, the model with the largest weight value is more prone to influence the final vote. Figure 7.5 presents the weighted majority voting implementation, within which weightings are applied to the base classifier outputs.

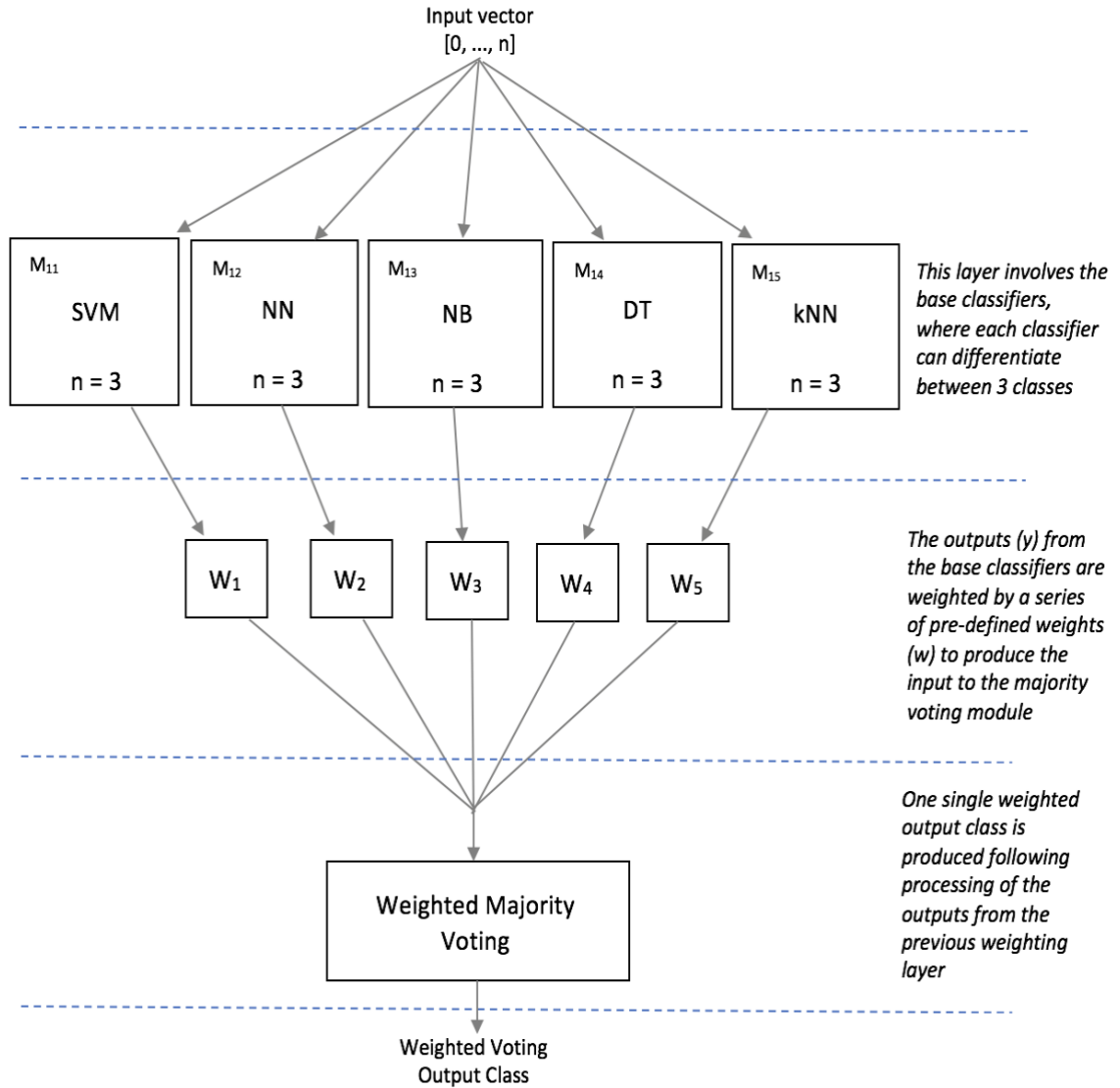


Figure 7.5. Weighted Majority Voting implementation where W represents the weights being applied to the outputs of each base classifier, and n represents the number of classes per model.

In Phase 2 of the proposed framework, the preliminary class decisions produced by each base model, M_1 to M_4 , were combined to integrate the outputs of the 4 time routines. Subsequently, conflict analysis was performed to ascertain

whether more than one model had chosen a main class output, resulting in a conflict requiring resolution. Table 7.2 presents an example of the combined outputs derived from each base model, demonstrating a conflict occurring between base models M_2 and M_3 , as M_2 had chosen output class 4 and M_3 had chosen output class 7 which were both main activity classes.

Table 7.2. Example of a conflict occurring between Base Models M_2 and M_3

		Base Model M ₁			Base Model M ₂		Base Model M ₃			Base Model M ₄							
		C ¹		C ¹	C ²	C ²	C ³		C ³	C ⁴							
Class label	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Chosen Class	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	

Following initial conflict analysis, conflict resolution subsequently occurred to decide upon which model would be awarded the final output class decision. As previously mentioned, the 4 methods used in the previously implemented homogeneous ensemble do not apply to the current work, as the combination techniques differed. In the previous approach, the output prediction values of each base classifier were used to compute a score per model through awarding the final decision to the model achieving the highest prediction value, the highest differential value, or through weighting the output prediction values by the no. of classes and training performances achieved. Since hard-level combination was implemented in Phase 1 of the current heterogeneous method, 5 newly constructed conflict resolution techniques were implemented to ascertain the final output decision based upon the following criteria: the strength of votes, the number of classes per base model, and the average training performance of the 5 classifiers per routine.

The first conflict resolution approach, presented in Algorithm 7.2.1, involved awarding the final output decision to the base output with the highest average training performance achieved through generating the 5 diverse base classifiers during Phase 1, as this is indicative of the predictive power of each base model (M_1 to M_4). The

second conflict resolution approach, Algorithm 7.2.2, involved awarding the final output decision to the base model comprising the largest number of classes as the higher the number of classes, the more complex the problem. An extension of the aforementioned methods was explored in the third approach, presented in Algorithm 7.2.3, within which both the average training performance achieved through generating the 5 diverse base classifiers and the number of classes per routine were considered, specifically the highest value produced by the average performances multiplied by the number of classes was awarded the final decision. Following this, the strength of the votes during ensemble integration in Phase 1 were considered in the fourth method, presented in Algorithm 7.2.4, where the average training performance achieved through generating the 5 diverse classifiers was multiplied by the strength of the vote. For example, considering the majority voting scenario, the strength was determined as the number of models producing the same vote. A vote strength of 4 was applied if 4 out of 5 classifiers had chosen the same output class, whereas a vote strength of 5 was applied if a unanimous class decision was reached across the suite of base classifiers. Finally, the last conflict resolution approach, presented in Algorithm 7.2.5, involved a combination of each criteria, within which the final class decision was awarded to the base model with the highest value attained through multiplying the average training performance achieved through generating the 5 diverse classifiers, the number of classes per base model, and the strength of the vote achieved during Phase 1.

Algorithm 7.2.1 Conflict resolution approach 1, Method 2.

Input: \vec{x}_i , base models M_r, M_s

Output: class y_i

- 1: $\overline{Acc}_{train}^r = \frac{\sum_{j=1}^5 Acc_{train}^{rj}}{5}; \overline{Acc}_{train}^s = \frac{\sum_{j=1}^5 Acc_{train}^{sj}}{5}$
 - 2: If $\overline{Acc}_{train}^r > \overline{Acc}_{train}^s$
 - 3: Then $y_i = \hat{k}_i^r$
 - 4: Else $y_i = \hat{k}_i^s$
-

where \overline{Acc}_{train}^r is the average training models of the 5 classifiers in Base Model M_r , and $Acc_{train}^{rj}, j = 1, \dots, 5$ represent the training performance of the model built using SVM, NN, NB, DT and kNN respectively in Base Model M_r .

Algorithm 7.2.2 Conflict resolution approach 2, Method 2.**Input:** \vec{x}_i , base models M_r, M_s **Output:** class y_i

- 1: $if\ m_r > m_s$
- 2: Then $y_i = \hat{k}_i^r$
- 3: Else $y_i = \hat{k}_i^s$

Algorithm 7.2.3 Conflict resolution approach 3, Method 2.**Input:** \vec{x}_i , base models M_r, M_s **Output:** class y_i

- 1: $if\ m_r \times \overline{Acc}_{train}^r > m_s \times \overline{Acc}_{train}^s$
- 2: Then $y_i = \hat{k}_i^r$
- 3: Else $y_i = \hat{k}_i^s$

Algorithm 7.2.4 Conflict resolution approach 4, Method 2.**Input:** \vec{x}_i , base models M_r, M_s **Output:** class y_i

- 1: $if\ w_r \times \overline{Acc}_{train}^r > w_s \times \overline{Acc}_{train}^s$
- 2: Then $y_i = \hat{k}_i^r$
- 3: Else $y_i = \hat{k}_i^s$

where w_r is the strength of the vote for the majority class for base model M_r defined by the number of classifiers with the majority class output in the base model.

Algorithm 7.2.5 Conflict resolution approach 5, Method 2.**Input:** \vec{x}_i , base models M_r, M_s **Output:** class y_i

- 1: $if\ m_r \times w_r \times \overline{Acc}_{train}^r > m_s \times w_s \times \overline{Acc}_{train}^s$
- 2: Then $y_i = \hat{k}_i^r$
- 3: Else $y_i = \hat{k}_i^s$

Figure 7.6 presents Phase 2 of the proposed heterogeneous approach in which the class decision outputs of each base model, M_1 to M_4 , generated during Phase 1 were combined through classifier fusion. Following this, conflict analysis and resolution occurred, and the performances of each conflict resolution approach were compared to ascertain the most effective technique.

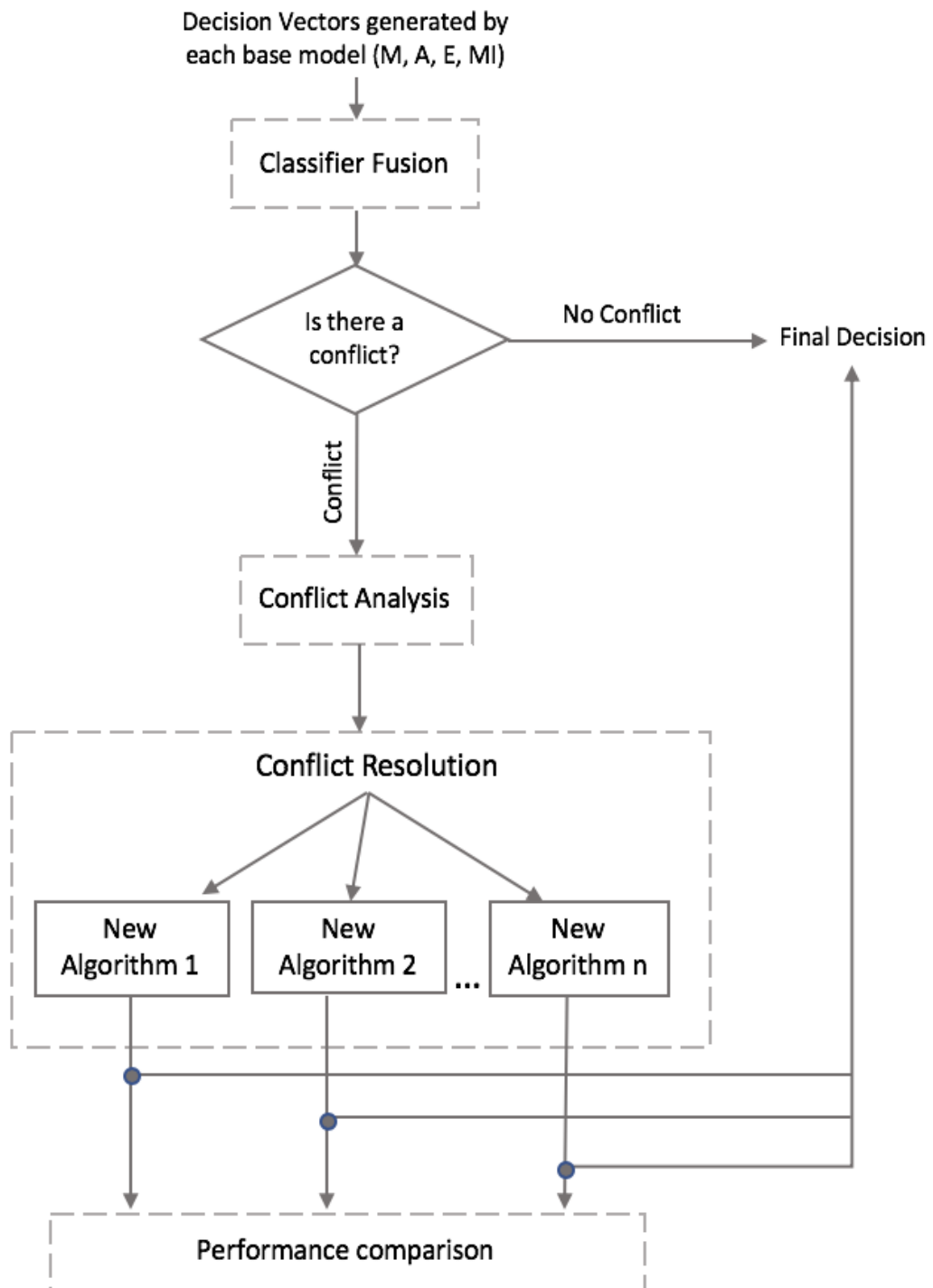


Figure 7.6. Phase 2 of the proposed heterogeneous approach, within which new conflict resolution algorithms were implemented.

7.4 Results and Discussion

The results obtained have demonstrated the effectiveness of heterogeneous ensemble methods. In terms of the number of conflicts occurring between base models, these were drastically reduced in comparison to the previously implemented homogeneous ensemble in Chapter 6, which contained 56.1 conflicting occurrences on average for the model-level complement class approach, and 23.3 on average for the class-level complement class approach. Table 7.3 presents the number of conflicts occurring per heterogeneous ensemble method. Method 1 proved most effective in minimizing the number of conflicts emerging, with only 9.7 on average, followed by the weighted majority voting approach in Method 2 with only 10.7 conflicts occurring on average. The low number of conflicts occurring within Method 1 was due to the output class decision of each base model relying solely upon the best performing classifier. As stated, the training performances achieved by each base classifier was indicative of their predictive power upon classifying the unseen test data. Thus, the decision to award the output class decision to the highest performing base classifier, whilst disregarding the remaining diverse classifiers proved beneficial in reducing the number of conflicts occurring. This evaluation is reinforced whilst considering the low number of conflicts occurring through applying the weighted majority voting method, as the weightings applied were based upon the training performances achieved by each diverse classifier.

Table 7.3. Conflicts occurring between base models

	No. of Conflicts Per Fold										Avg.
	1	2	3	4	5	6	7	8	9	10	
Method 1	6	19	10	10	12	11	6	9	6	8	9.7
Method 2 via Majority Voting	10	19	12	8	27	20	17	24	17	21	17.5
Method 2 via Weighted Majority Voting	6	13	6	8	23	14	5	20	4	8	10.7

Classification performances were evaluated before conflict analysis occurred to obtain the preliminary performance per method, and subsequently evaluated following the implementation of each conflict resolution method explored to determine their effectiveness. Figure 7.7 presents the HAR performances achieved through the implementation of heterogeneous ensemble method 1. Due to the predictive power of the chosen diverse classifiers and low number of conflicts occurring, method 1 performed reasonably well before conflict analysis and resolution, achieving a preliminary classification accuracy of 78.60%. Following this, the final performances achieved through the implementation of conflict resolution algorithms 7.1.1 to 7.1.3 resulted in classification accuracies of 80.84%, 80.45% and 80.45%, respectively.

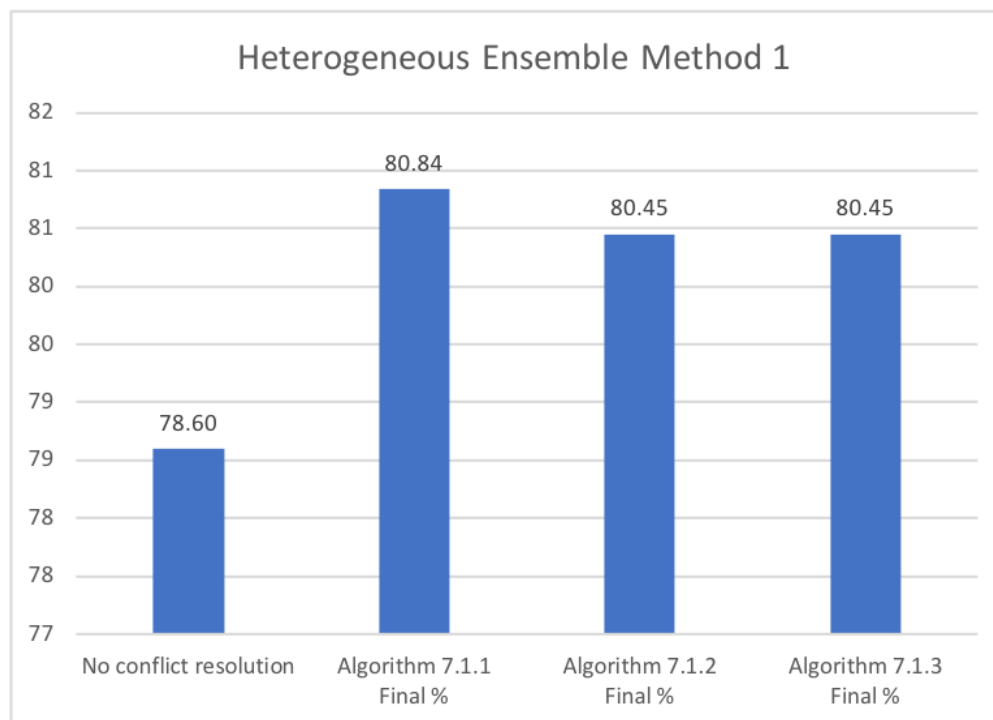


Figure 7.7. HAR performances achieved through Heterogeneous Ensemble Method 1

Figure 7.8 presents the classification performances achieved through heterogeneous ensemble method 2. The preliminary performances achieved through majority voting and weighted majority voting during Phase 1 were 77.88% and 79.89%, respectively. This demonstrated the effectiveness of applying additional

weighting to votes through also considering the average training performances achieved by each diverse classifier during training. Following the implementation of Phase 2, which involved combining the outputs from each base model, M_1 to M_4 , an optimal final classification accuracy of 84.13% was achieved through applying conflict resolution algorithm 7.2.4, previously described in Section 7.3.2. Furthermore, the next best performing method involved applying the same conflict resolution technique, algorithm 7.2.4, in conjunction with Method 2 via majority voting. Thus, demonstrating the effectiveness of considering both the average training performances and the strength of the votes provided by each base classifier.

The least effective conflict resolution approach was that of algorithm 7.2.1, previously described in Section 7.3.2, attaining 81.34% accuracy in conjunction with majority voting technique, and 83.13% accuracy in conjunction with the weighted majority voting method. This demonstrated that during Phase 2, considering training performance alone was insufficient when using algorithm 7.2.1 in conjunction with the majority voting technique, and additional support was required through factoring in further information to obtain optimal performance. For example, considering only the training performances at this stage resulted in models M_1 to M_3 attaining advantage in comparison to M_4 , as they achieved much higher classification accuracies due to their more simplistic classification task in that they each comprised a low number of classes. The classification task conducted by model M_4 was more intricate due to the larger number of classes involved, yet, this model would have been disregarded during conflict resolution whilst in resolve against any of the remaining models.

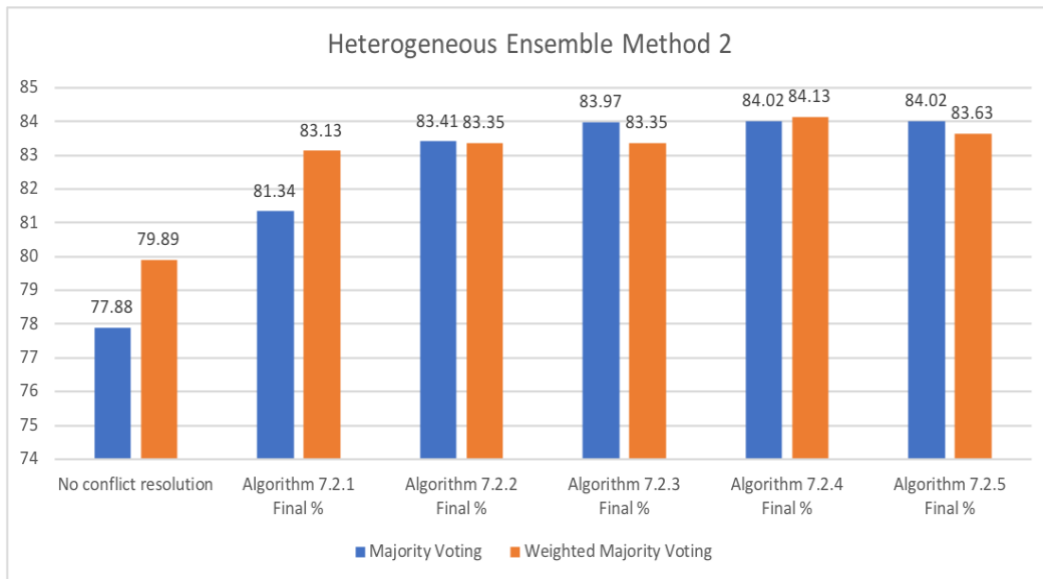


Figure 7.8. HAR performances achieved through Heterogeneous Ensemble Method 2

Tables 7.4, 7.5 and 7.6 present an analysis of incorrect instances pertaining to each implemented heterogeneous method. The “incorrect” instances represent those that were incorrectly classified, regardless of any combination techniques or conflict resolution, as these describe misclassified instances by the target model which are important to reflect upon in evaluating the most effective conflict resolution methods. The “right but incorrect” instances represent those that were classified accurately by the target model, yet were disregarded due to the application of conflict resolution approaches.

Table 7.4. Ensemble method 1 - Analysis of incorrect instances

		Fold										Avg.
		1	2	3	4	5	6	7	8	9	10	
A 7.1.1	Incorrect	29	37	36	33	37	36	35	35	30	35	34.3
	Right but Incorrect	3	12	6	6	9	7	2	5	2	5	5.7
A 7.1.2	Incorrect	30	34	38	35	35	38	37	38	32	33	35.0
	Right but Incorrect	4	9	8	8	7	9	4	8	4	3	6.4
A 7.1.3	Incorrect	30	34	38	35	35	38	37	38	32	33	35.0
	Right but Incorrect	4	9	8	8	7	9	4	8	4	3	6.4

Through analysis of Table 7.4, conflict resolution algorithm 7.1.1 was deemed the optimally performing technique, applied during Method 1. This was due to algorithm 7.1.1 resulting in the least number of “right but incorrect” cases (5.7 on average) in comparison to the remaining evaluated techniques. Nevertheless, algorithms 7.1.2 and 7.1.3 closely followed this by equally resulting in 6.4 “right but incorrect” cases, on average. The average number of conflicts that had occurred with algorithm 7.1.1 were then compared to the pertaining “right but incorrect” cases to ascertain the number of instances that were correctly resolved as a result of applying this technique. For example, there were 9.7 base model conflicts on average, and through application of algorithm 7.1.1 there were an average of 34.3 incorrect instances. Yet, an average of 5.7 of those could have been resolved through an efficient resolution method. Consequently, the final HAR performance achieved was enhanced by an average of 4 instances.

Subsequently, the conflict resolution techniques explored during Method 2 were evaluated through analysis of Tables 7.5 and 7.6. The most effective conflict resolution technique was algorithm 7.2.4 in conjunction with the weighted majority voting method, as a low average of 4.4 instances were deemed “right but incorrect”. The number of occurring conflicts on average during this technique were then compared to the associated “right but incorrect” cases to determine the number of instances that were correctly resolved. An average of 10.7 conflicts emerged on average, and through applying algorithm 7.2.4, an average of 28.4 incorrect instances transpired. Thus, considering an average of 4.4 instances may have been resolved, the final HAR performance was consequently enhanced by 6.3 instances, on average.

Algorithm 7.2.5 in conjunction with the weighted majority voting method was deemed the next best performing approach, as an average of 5.3 instances were deemed “right but incorrect”. Furthermore, the least effective conflict resolution approach was that of Algorithm 7.2.1 which attained an average of 11.3 “right but incorrect” cases. As previously mentioned, this demonstrated that relying on training performance alone was inadequate during Phase 2, when used in conjunction with majority voting.

Table 7.5. Ensemble method 2 via majority voting – Analysis of incorrect instances

		Fold										Avg.
		1	2	3	4	5	6	7	8	9	10	
A 7.2.1	Incorrect	32	33	36	31	35	35	31	32	34	35	33.4
	Right but Incorrect	5	11	8	4	19	14	11	15	12	14	11.3
A 7.2.2	Incorrect	32	31	32	31	27	29	28	28	29	30	29.7
	Right but Incorrect	5	9	4	4	11	8	8	11	7	9	7.6
A 7.2.3	Incorrect	31	30	31	30	26	28	27	27	28	29	28.7
	Right but Incorrect	4	8	3	3	10	7	7	10	6	8	6.6
A 7.2.4	Incorrect	28	29	30	28	27	29	26	29	31	29	28.6
	Right but Incorrect	1	7	2	1	11	8	6	12	9	8	6.5
A 7.2.5	Incorrect	31	30	31	30	26	27	27	27	28	29	28.6
	Right but Incorrect	4	8	3	3	10	6	7	10	6	8	6.5

Table 7.6. Ensemble method 2 via weighted majority voting – Analysis of incorrect instances

		Fold										Avg.
		1	2	3	4	5	6	7	8	9	10	
A 7.2.1	Incorrect	29	30	31	32	32	30	28	30	29	31	30.2
	Right but Incorrect	2	8	3	5	16	9	2	13	1	3	6.2
A 7.2.2	Incorrect	31	28	31	34	26	28	29	27	31	33	29.8
	Right but Incorrect	4	6	3	7	10	7	3	10	3	5	5.8
A 7.2.3	Incorrect	31	31	31	34	26	28	29	27	31	33	30.1
	Right but Incorrect	4	11	3	7	10	7	3	10	3	5	6.3
A 7.2.4	Incorrect	28	30	28	28	28	27	26	30	29	30	28.4
	Right but Incorrect	1	8	0	1	12	6	0	13	1	2	4.4
A 7.2.5	Incorrect	31	28	31	30	26	27	29	27	31	33	29.3
	Right but Incorrect	4	6	3	3	10	6	3	10	3	5	5.3

In Figure 7.9, comparisons of each ensemble method were made in terms of the final classification accuracies achieved following implementation. Both heterogeneous ensemble methods explored outperformed the homogeneous approach, demonstrating the effectiveness of generating further diversity amongst various classifiers. Thus, the obtained results indicate generating diversity at both data and classifier levels has proven more effective than solely generating diversity at a data level. According to [200], this is due to diverse base classifiers providing additional benefits through offering different biases and internal representations. Significance testing was applied through T-testing with a 95% confidence to ascertain whether the obtained results were statistically significant during performance comparisons amongst the homogeneous and heterogeneous ensemble classifiers presented in Figure 7.9. Thus, a p -value of <0.05 was deemed significant. Results demonstrated insignificance when comparing the homogeneous ensemble to heterogeneous method 1 (p -value of 0.554739455), however, statistical significance was achieved whilst comparing the homogeneous ensemble to heterogeneous method 2 (p -value of 0.00000759158).

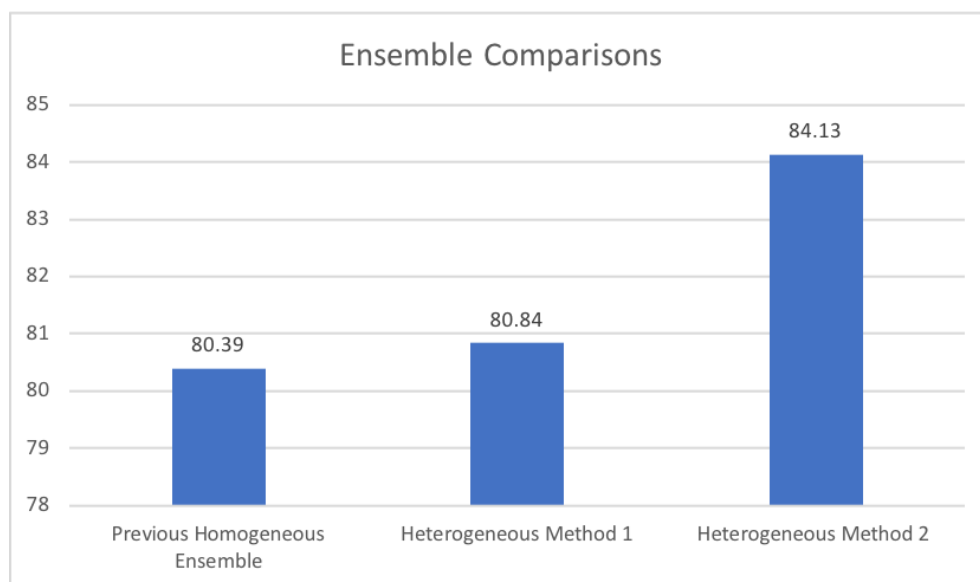


Figure 7.9. Comparisons of each ensemble method, including the homogeneous method, and heterogeneous methods 1 & 2

The best performing heterogeneous ensemble method was that of the 2-phase technique, Method 2 via weighted majority voting, which achieved a final classification accuracy of 84.13%. This demonstrated an accuracy increase of 3.74% in comparison to the homogeneous ensemble, and an increase of 3.29% in comparison to heterogeneous ensemble method 1. Through interpretation of the obtained heterogeneous results within this Chapter, Method 2 outperformed Method 1 due to the enhanced predictive power provided by retaining valuable information offered by each of the 5 diverse classifiers, through voting. Method 1 involved comparing the training performances achieved by each diverse classifier, per time routine, to ascertain which of the 5 would be selected to represent the base model. The classifier achieving the highest training performance was retained, whereas the remaining 4 classifiers were disregarded. A limitation of this approach was identified in that potentially valuable information may have been provided by the remaining 4 classifiers. Consequently, Method 2 was proposed to overcome this limitation, through combining the output predictions of all 5 diverse classifiers via voting. This approach subsequently had the effect of strengthening the predictions which were obtained and retaining all potentially valuable information. The heterogeneous ensemble results have supported this evaluation. Notably, the heterogeneous ensemble Method 2 results had also successfully outperformed all benchmark classifiers defined in previous Chapters, namely the kNN, NN, SVM and LR models which had attained accuracies of 70.95%, 80.45%, 76.54% and 81.01%, respectively. Further to this, heterogeneous ensemble Method 1 successfully outperformed all 3 non-parametric benchmark classifiers, whilst the remaining classifier had demonstrated a very slight performance increase of 0.17%. Thus overall, largely demonstrating the effectiveness of heterogeneous ensembles, and particularly demonstrating the comprehensive superiority of the proposed heterogeneous Method 2.

7.5 Conclusion

This Chapter provided further exploration upon ensemble methods initially introduced within Chapter 6. In Chapter 6, homogeneous ensembles were the focus, whereas the opportunity to explore heterogeneous ensembles has been exploited

within the current Chapter. As previously mentioned, it has been recognised that achieving diversity may transpire through either altering the training data and/or base classifiers, however, no clear consensus exists in defining or measuring diversity [51]. In the previously implemented homogeneous ensemble, diversity was explored at a data level only through diversifying the inputs of each NN base classifier. Within the current work, diversity was explored at both a data level and a classifier level through additionally generating diverse classifiers, namely an SVM, NB, NN, kNN and DT, as the literature suggested heterogeneous ensembles may provide additional benefits through diversifying the classifiers employed [130], [200].

Two methods of generating heterogeneous ensembles were explored. Method 1 involved generating 5 base classifiers per time routine, within which the best performing classifier in terms of accuracy was selected to represent the considered base model, M_1 to M_4 . Consequently, an SVM represented the M_1 model, an NB represented the M_2 model, an NN represented the M_3 model and a DT represented the M_4 model. Following this, the output class predictions were combined, and finally, conflict resolution approaches were applied and evaluated. Contrarily, Method 2 involved a two-phase approach within which two levels of ensemble integration were explored. Within Phase 1, both majority voting and weighted majority voting were applied to combine the output class predictions of all 5 diverse classifiers, per time routine, and were then subsequently compared. Within Phase 2, the resultant output class predictions were combined, and lastly, conflict resolution techniques were applied and evaluated.

The experimental results obtained through the exploration of heterogeneous ensembles demonstrated their effectiveness in comparison to the homogeneous method in Chapter 6, indicating that introducing additional diversity was an effective approach. Additionally, heterogeneous method 2 outperformed method 1, with algorithm 7.2.4 proving the most effective conflict resolution approach in conjunction with the weighted majority voting method. Algorithm 7.2.4 involved awarding the final decision to the model with the highest value attained through multiplying the average training performance of the 5 diverse base classifiers by the strength of the votes within each conflicting base model, which ultimately proved most superior.

Chapter 8

Conclusions and Future Work

8.1 Overview

This Chapter discusses the key findings and conclusions generated as a result of the research conducted throughout this Thesis. The overarching research endeavour of this Thesis intended on exploring methods to enhance HAR performance within smart environments. This endeavour was achieved through establishing an end-to-end methodology for the optimisation of HAR which focused on enhancing HAR performance at various stages of the process, from data acquisition through to activity classification. Upon reflection, the end-to-end

methodology consisted of 3 main components, specifically pre-processing, feature extraction, and classification. Performance enhancements were demonstrated within each component, however, within the classification stage the developed homogeneous ensemble did not perform as well as had been anticipated. Nevertheless, the developed heterogeneous ensemble classifier demonstrated notable success.

A comprehensive literature review was initially conducted to assess the current state of the art and to identify research opportunities/future trends pertaining to HAR within smart environments. This review considered the 5 stage activity recognition process, involving data acquisition, pre-processing, segmentation, feature extraction and selection, and classification, as well as considering various application domains for HAR research and identifying predominant research challenges associated with HAR. Consequently, research studies were designed and conducted throughout this Thesis to enhance HAR performance within smart environments at each stage of the process, with particular focus and contributions upon pre-processing, feature extraction and selection and classification.

Section 8.2 presents a discussion of the work, Section 8.3 presents a summary of the key research contributions, Section 8.4 outlines identified limitations of the research studies conducted and Section 8.5 proposes future work and highlights future research directions. Finally, Section 8.6 ultimately concludes this Thesis.

8.2 Discussion

The fundamental aim of this Thesis was to explore methods of enhancing HAR performance within smart environments. This has been achieved through establishing an end-to-end methodology for the optimisation of HAR which focused on enhancing HAR performance at various stages of the process, from data acquisition through to activity classification.

The following research questions were defined to help achieve the aim of this Thesis and have been addressed through reviewing the literature and conducting research studies:

1. What are the research challenges associated with HAR that may hinder classification performance?

2. To what extent does data quality impact upon HAR performance?
3. Can hybrid feature selection methods offer additional benefits in producing an optimal subset of features for HAR?
4. Can generating diversity within ensemble learners effectively enhance HAR performance?

The first research question was considered within Chapter 2. A critical challenge that continues to hinder HAR performance has been identified as the lack of available, high quality datasets to evaluate research studies [13]. Furthermore, the challenges of class imbalance, interclass similarity, intraclass variability and recognising interleaved and concurrent activities were identified and discussed.

The second research question was investigated within Chapter 3 through conducting a study where HAR performance was evaluated before, and after, data cleaning to ascertain the impact of noise upon classification, conclusively demonstrating the adverse impact of noise.

Following this, the third research question was investigated within Chapter 5 through exploring a suite of well-established feature selection methods, in addition to developing a new hybrid feature selection method which demonstrated its effectiveness and benefits in comparison to the initially explored techniques.

Finally, the fourth research question was investigated within Chapters 6 and 7 which demonstrated the performance enhancing capabilities of ensemble classifiers by developing novel homogeneous and heterogeneous classifiers. Particularly, the experimental results obtained through the exploration of heterogeneous ensembles demonstrated their effectiveness in comparison to the homogeneous method, indicating that introducing additional diversity was an effective approach.

Generally, sensor-based HAR is approached through implementing either data-driven or knowledge-driven classifiers. Research studies conducted within this Thesis focussed upon data-driven classification. Nevertheless, knowledge has been inferred throughout various stages of the end-to-end methodology which enhanced the quality of inputs to each data-driven classifier generated. For example, the data restructuring process within Chapter 4 was driven through knowledge of the available sensors and the activities to be classified. Several challenges were identified, such as the large quantity of activity classes with regards to the low amount of information provided by binary sensors at an abstract level. It was recognised that certain

activities were indistinguishable with the available sensors, such as Act21 work at the table, and certain activities were too complex to recognise with binary sensors alone. For example, Act10 enter smart lab, Act13 leave smart lab and Act14 visitor to smart lab were indistinguishable given that only one binary sensor was available at the smart lab doorway region. Further to this, knowledge was also inferred at the feature selection stage through analysing the initial results and discovering common features chosen for removal by all filter methods, in addition to observing the differences in performances achieved by each filter method and deliberating whether a truly optimal feature subset had been discovered. Consequently, a new hybrid feature selection method was developed. Finally, knowledge was also inferred at the classification stage, in that each base model within the ensemble classifiers were organised according to which activities habitually occurred within each time routine. Thus, knowledge of daily routines was inferred as humans habitually get up in the morning, prepare a meal in the morning, afternoon and evening, and also go to bed in the evening, for example.

It is recognised that a solely data-driven approach would be capable of providing more valuable information given an additional range of sensor features. For example, the inclusion of a binary sensor in the table region would support the recognition of Act21 work at the table, or perhaps more sensor modalities such as the inclusion of proximity data to distinguish between Act10 enter smart lab, Act13 leave smart lab. Nevertheless, a solely data-driven approach is not recommended with the original data containing 24 classes as certain activities could not be distinguished with binary sensors alone, and the original data was highly imbalanced. Instead, the data was restructured based on inferred knowledge to reveal the potential of binary sensors within a smart environment setting. Another interesting approach could have involved recapturing the activity data with the inclusion of additional sensors to enhance the level of information provided per activity class, thus also improving the end-to-end methodology.

8.3 Summary of Contributions

An end-to-end methodology was established within this thesis to ascertain means in which HAR performance could be enhanced at various stages of the HAR

process to achieve the overarching research aim of this Thesis of improving the performance of HAR within smart environments. A publicly available HAR dataset containing data streams of several common ADLs was taken through the proposed end-to-end methodology, with particular focus upon exploring methods to enhance performance. Through conducting a comprehensive literature review of the HAR process, various promising opportunities of enhancing performance within each stage were identified and explored, rather than solely focussing on enhancing performance within the classification stage alone. It was recognised that whilst various research endeavours within the HAR community focus upon enhancing overall performance at the classification stage alone, the overall HAR performance within smart environments could be enhanced incrementally by investigating various stages of the process. Figure 8.1 highlights the HAR stages explored within this Thesis, which included data cleaning, scrubbing and wrangling, feature extraction and selection, and constructing a classification model.

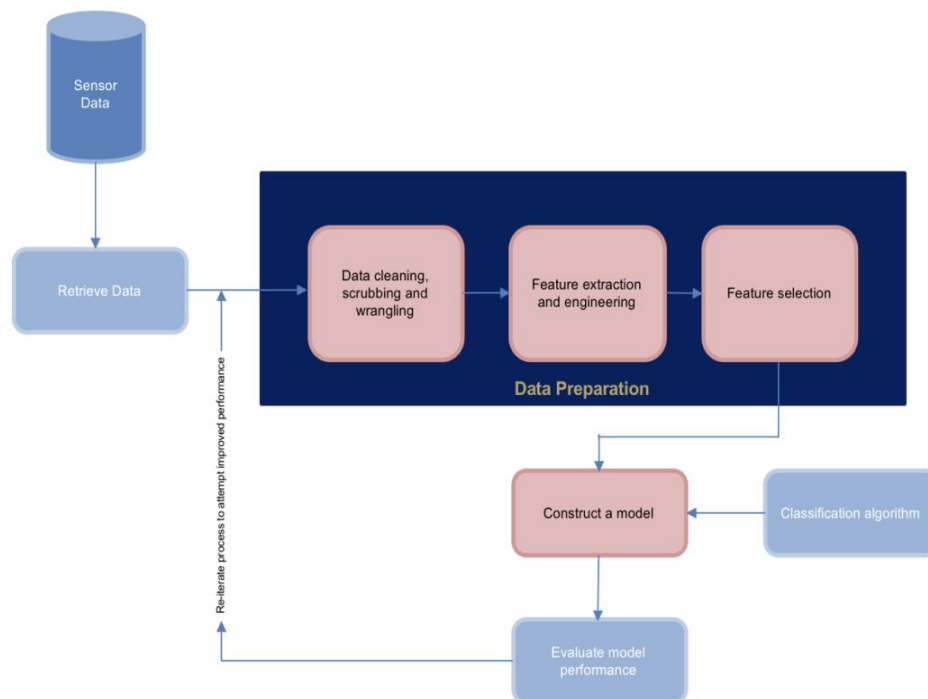


Figure 8.1. The HAR process, adapted from [16], which highlights, in red, the key areas explored within this Thesis

This thesis has provided the following contributions:

- **Recommendations for pre-processing data to improve the performance of data-driven approaches to HAR**

Chapter 3 presented an experimental study which was conducted to explore the impact of dataset quality upon classification using data-driven models. This study was based on the premise that data quality is a crucial consideration for the development of data-driven classification models. An endeavour to generate performance comparisons between noisy, and subsequently cleaned data, was presented within Chapter 3. The methodology undertaken to clean the data demonstrated its effectiveness as results revealed performance enhancements during classification with the entire suite of classifiers evaluated, namely a NN, DT, kNN and NB, through discovering and removing noisy portions of data.

Findings from this work have proven the benefits of generating classifiers with cleaned data, and also suggested that the presence of noise in accelerometry data has a negative impact on the performance of data-driven classifiers, thus supporting their need for high quality data. This subsequently led to the production of a set of guidelines of how to clean accelerometry data for the purposes of HAR. This work highlighted the importance of adhering attentively to a data collection protocol to minimise the introduction of noise during data acquisition, and recommended that data should be initially screened to verify its quality for classification purposes. Experimental findings also lead to a recommendation of further refining the data collection protocol to minimise the introduction of noise, as the existence of noise was predominantly due to failure in following the protocol.

Chapter 4 involved identifying data challenges within a publicly available HAR dataset and provided recommendations as to how these challenges should be solved. Findings from both Chapters 3 and 4 reinforced the need to develop refined study protocols and standards for the collection and storage of datasets to ensure the effective collection and dissemination of high quality HAR datasets. Furthermore, the produced recommendations can be used as guidelines for supporting data collection protocols, which will ensure the collection of good quality data and also encourage data sharing.

- **Produced a new approach to select an optimal subset of features for HAR**

Chapter 5 presented the details of a research study conducted to discover an optimal subset of features for HAR within smart environments using environmentally deployed binary sensors. Following initial experiments with conventional techniques, the opportunity emerged to implement a hybrid-filter technique to further enhance classification performance as the literature had indicated the potential of hybrid methods to further enhance performance. Thus, Chapter 5 presented a novel hybrid-filter feature selection method, which generated new subsets of features for removal. The hybrid-filter demonstrated enhanced HAR performance and revealed a considerable trade-off between the classification performances achieved and the number of redundant features identified and removed, in comparison to all initially evaluated conventional techniques. Experimental results demonstrated that reducing data dimensionality through removing redundant features, had either maintained or enhanced HAR performance amongst the full suite of evaluated classifiers. The advantages of combining feature selection methods were also demonstrated through implementing the proposed hybrid-filter, as the combined filtering methods had complemented each other to achieve an ultimately superior subset of features for ADL recognition.

- **A new homogeneous ensemble classification model that introduces diversity at a data level**

Chapter 6 presented a research study conducted to enhance HAR classification performance by exploring ensemble methods due to their perceived benefits over individual models. A novel approach to generating data diversity was presented within Chapter 6, within which 4 NN base models were generated to represent activities per time routine, specifically the Morning, Afternoon, Evening and Mixed routines. The Mixed model was generated to represent activities that habitually occur throughout a typical daily routine. During the ensemble generation stage, explorations were made to analyse the effects of

various complement class data compositions amongst the base models as each model had been trained with unique activity classes.

Further to this, various conflict resolution techniques were designed to resolve conflicts occurring between base models during the ensemble integration stage. The designed techniques included either awarding the final output decision to the model with the largest probability output prediction, the largest differential output value between the conflicting models, or weighting techniques using the training performances attained and the number of classes per conflicting model. The most effective conflict resolution technique was that of Algorithm 6.3, which involved awarding the final output decision to the model with the largest differential value between the conflicting models, as this approach resulted in the least number of “right but incorrect” instances when applied to both data distributions.

The performance of the proposed homogeneous NN ensemble was benchmarked against the suite of classifiers introduced in Chapter 5, namely kNN, SVM, NN and LR. The proposed ensemble outperformed 2 from the 4 classifiers considered, namely the kNN and SVM models, and was very slightly less effective than the NN and LR models, which outperformed the proposed approach by 0.06% and 0.62%, respectively.

- **A new heterogeneous ensemble classification model that introduces diversity at both data and classifier levels**

Chapter 7 presented a research study following the work introduced in Chapter 6 by further enhancing the level of diversity within an ensemble method, thus endeavouring to further enhance HAR performance. Within Chapter 7, a novel heterogeneous ensemble classifier was designed. Diversity was explored at both data and classifier levels, utilising the previously explored data diversity method, in addition to diversifying the base classifiers through exploiting the benefits of a number of diverse classifiers, namely kNN, NB, SVM, NN and DT models, as these models were identified as complementary within previous works [200].

Two heterogeneous ensemble generation methods were explored. Method 1 involved generating 5 base classifiers per base model M_1 to M_4 (representing the

Morning, Afternoon, Evening and Mixed routines, respectively), where the classifier achieving the highest training performance was chosen to represent each base model. Contrarily, Method 2 involved a two-phase ensemble integration approach, in which voting schemes, namely majority voting and weighted majority voting, were implemented and compared within Phase 1 to combine the outputs from each of the diverse classifiers, and subsequently the resulting base model decisions were combined, per base model M_1 to M_4 , in Phase 2.

Conflict resolution techniques were applied to both Method 1 and Method 2 to combine the outputs of each base model. Within Method 1, there were 3 suitable conflict resolution techniques explored, whereas within Method 2, the opportunity of exploring 5 conflict resolution techniques had emerged. The optimally performing heterogeneous ensemble method was that of Method 2 via weighted majority voting, in conjunction with conflict resolution algorithm 7.2.4, which involved awarding the final output decision to the model with the highest value through multiplying the average training performance achieved through generating the 5 diverse classifiers, by the strength of the vote attained. Both heterogeneous ensemble methods outperformed the homogeneous NN ensemble proposed in Chapter 6, in addition to the heterogeneous Method 2 outperforming all other benchmark classifiers considered within previous Chapters, namely the kNN, SVM, NN and LR classifiers by 13.18%, 7.59%, 3.68% and 3.12%, respectively, which had ultimately proven its superiority.

8.4 Limitations

One sensor-based HAR dataset has been taken through the proposed end-to-end framework with thorough evaluations performed at each stage of the process. A limitation of this Thesis has been identified in that only one HAR dataset was utilised to evaluate the end-to-end framework. It is acknowledged that the inclusion of another dataset would further support the findings produced through conducting experimental studies. Nevertheless, the low availability of good quality, accurately annotated HAR datasets remains a challenge, as recognised in Chapters 3 and 4.

Another identified limitation is that the optimal feature subset discovered within Chapter 5 was not progressed into the succeeding Chapters of the end-to-end methodology. The experimental work conducted within Chapter 5 was retrospective to the succeeding Chapters, thus the full feature set was used as input to each base classifier within the ensemble Chapters.

It is appreciated that in the domain of smart environments, a number of data types are available. Within Chapter 3, continuous data was evaluated whereas binary data was considered within Chapters 4-7. A limitation is identified in that the conclusions derived from the research study evaluating continuous data may not be generally applicable to binary data, and vice versa. It is unclear whether the same, or similar, conclusions would be generated. Thus, further experimentation would be required to ascertain whether the conclusions generated are transferrable amongst both data types.

It is recognised that several performance metrics exist and are employed for various classification tasks. The accuracy and F-measure metrics have been utilised in various HAR studies [13], [85], [106], [203], [204], demonstrating their common employment. Within the studies conducted in this thesis, accuracy was chosen as the sole performance metric as the utilised data was originally generated for the UCAmI Cup competition, which had specified accuracy as the sole performance metric to evaluate the participants' proposed methods. Thus, the decision upon choice of evaluation metric was influenced by the UCAmI Cup guidelines to facilitate comparison with other approaches using the same data. Nonetheless, upon reflection, it is acknowledged that the inclusion of additional performance metrics may have been beneficial in evaluating the proposed methods. For example, due to the common existence of class imbalance within HAR datasets, it has since been recognised that including the F-measure metric could provide additional benefit in that it presents an unbiased measure of accuracy in imbalanced environments by combining precision and recall scores [204].

8.5 Proposed Future Work

Various areas demonstrating potential for further exploration have been identified to extend the contributions made within this Thesis:

Given that Chapters 3 and 4 emphasised the importance of data quality for data-driven HAR and highlighted that data collection for HAR purposes is becoming a critical challenge within this domain, it has been recognised that clear data collection and dissemination standards should be developed for researchers to effectively evaluate their research. Data collection protocols should adhere to the developed standards to minimise the introduction of noise and avoid the production of suboptimal quality datasets that adversely affect HAR performance. Providing such standards would encourage researchers to effectively collect and disseminate high quality datasets, thus eliminating the current challenge of HAR research being hindered by the low availability of publicly available datasets that include a large quantity of accurately annotated and high quality data. Further to this, it has also been recognised through findings within Chapter 4 that the quality of publicly available datasets is unclear [81], with many researchers expressing concerns pertaining to the quality of the UCAMi Cup dataset, for example. It was also found that data quality has a significant impact on the performance of data-driven approaches to HAR. Thus, a clear data quality assessment process should be established for researchers to benefit from, in that each publicly disseminated HAR dataset is labelled with a single value, based upon identified assessment criteria, which represents the overall quality of each HAR dataset.

Given that the results obtained within Chapter 5 were not utilised within succeeding Chapters, future work should investigate the impact of the optimally selected features discovered within Chapter 5 upon the classification methods explored within Chapters 6 and 7. This could potentially further enhance overall classification performance as the benefits of applying feature selection upon performance have been demonstrated through exploring several feature selection techniques.

Chapter 6 investigated various conflict resolution techniques to combine the homogeneous NN base models. The best performing technique was that of Algorithm 6.3, in which the highest differential value between the highest and second highest predictions, per conflicting base model, was awarded the final output decision, as this was deemed the model with the strongest class prediction. An opportunity to investigate an adaption of this algorithm exists, in that it would be interesting to consider this method only in the occurrence that the highest and second highest values exist within a certain range.

Chapters 6 and 7 investigated ensemble methods for HAR. Diversity was explored at a data level only in Chapter 6 through the implementation of a homogeneous method, and diversity was explored at both data and classifier levels in Chapter 7 through the implementation of a heterogeneous method. The heterogeneous ensemble method in Chapter 7 ultimately proved most superior, nevertheless, it would be interesting to further explore ensembles by developing a heterogeneous ensemble attaining diversity at a classifier level only to have explored diversity at all levels. Further to this, means of optimising the computational complexity of the models developed in Chapters 6 and 7 would be explored to reduce the computation time required and increase efficiency. For example, the chosen number of hidden layers could be adjusted whilst monitoring any potential impact upon performance.

Finally, the developed ensemble classifiers within Chapters 6 and 7 should be evaluated with another dataset as this would provide additional benefit in that the findings would be further supported and reinforced, in addition to further generalising the findings obtained. Particularly, a dataset with a larger quantity of high-quality data would be beneficial as providing more labelled training data could enhance the prediction quality of data-driven classification techniques.

8.6 Conclusion and Future Direction

This Chapter has concluded this Thesis through providing a summary of each of the research contributions made, outlining limitations encountered, and finally, outlining areas for potential future work. It has been recognised that significant advancements in sensor technology and wireless sensor networks have been made in recent years, which have supported the progression of HAR [14]. Nevertheless, the widely acknowledged lack of available data is continuing to hinder the development of (data-driven) HAR research [13]. Data collection remains a critical challenge obstructing the progression of activity recognition solutions within smart environments. Thus, an increase in shared data resources would vastly facilitate HAR research and particularly support the development and evaluation of HAR technologies within smart environments. Furthermore, there is benefit in enhancing other stages of the HAR process, which have been demonstrated within this Thesis.

References

- [1] WHO, “Ageing,” *World Health Organisation*, 2020. [Online]. Available: <https://www.who.int/health-topics/ageing#>. [Accessed: 18-Nov-2020].
- [2] W. H. Organisation, “Global Strategy and Action Plan on Ageing and Health,” 2017.
- [3] S. Wang *et al.*, “Technology to Support Aging in Place: Older Adults’ Perspectives,” *Healthcare*, vol. 7, no. 2, p. 60, 2019.
- [4] E.-M. Schomakers, J. O. Heek, and M. Ziefle, “Attitudes Towards Aging and the Acceptance of ICT for Aging in Place,” in *Nature*, Springer International Publishing, 2018, pp. 149–169.
- [5] L. Chen and C. D. Nugent, “Sensor-Based Activity Recognition Review,” in *Human Activity Recognition and Behaviour Analysis*, Springer Nature Switzerland AG, 2019, pp. 23–47.
- [6] J. K. Aggarwal, L. Xia, O. C. Ann, and L. B. Theng, “Human activity recognition: A review,” in *IEEE International Conference on Control System, Computing and Engineering*, 2014, pp. 389–393.
- [7] S. Ranasinghe, F. Al Machot, and H. C. Mayr, “A review on applications of activity recognition systems with regard to performance and evaluation,” *Int. J. Distrib. Sens. Networks*, vol. 12, no. 8, 2016.
- [8] E. Kim, S. Helal, and D. Cook, “Human Activity Recognition and Pattern Discovery,” *IEEE Pervasive Comput.*, vol. 9, no. 1, pp. 48–53, 2010.
- [9] N. C. Krishnan and D. J. Cook, “Activity Recognition on Streaming Sensor Data,” *Pervasive Mob. Comput.*, vol. 10, no. PART B, pp. 138–154, 2014.
- [10] M. E. Mlinac and M. C. Feng, “Assessment of Activities of Daily Living, Self-Care, and Independence,” *Arch. Clin. Neuropsychol.*, vol. 31, no. 6, pp. 506–516, Sep. 2016.
- [11] S. Yan, Y. Liao, X. Feng, and Y. Liu, “Real Time Activity Recognition on Streaming Sensor Data for Smart Environments,” in *IEEE International Conference on Progress in Informatics and Computing*, 2016, pp. 51–55.
- [12] X. Fan *et al.*, “Activity Recognition as a Service for Smart Home: Ambient Assisted Living Application via Sensing Home,” in *IEEE 6th International Conference on AI and Mobile Services*, 2017, pp. 54–61.

- [13] I. Cleland, M. P. Donnelly, C. D. Nugent, J. Hallberg, and M. Espinilla, "Collection of a Diverse , Naturalistic and Annotated Dataset for Wearable Activity Recognition," in *PerCom*, 2018.
- [14] L. Minh Dang, K. Min, H. Wang, M. Jalil Piran, C. Hee Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognit.*, vol. 108, 2020.
- [15] S. Todd, "Data Wrangling vs. Data Cleaning: What's the Difference?," 2020. [Online]. Available: <https://www.inzata.com/data-wrangling-vs-data-cleaning-whats-the-difference/>.
- [16] E. C. Dinarević, J. B. Husić, and S. Baraković, "Issues of Human Activity Recognition in Healthcare," in *18th International Symposium INFOTEH-JAHORINA, INFOTEH 2019*, 2019, no. March, pp. 20–22.
- [17] X. Su, H. Tong, and P. Ji, "Activity Recognition with Smartphone Sensors," *Tsinghua Sci. Technol.*, vol. 19, no. 3, pp. 235–249, 2014.
- [18] S. Gupta and A. Gupta, "Dealing with noise problem in machine learning datasets: A systematic review," *Procedia Comput. Sci.*, vol. 161, pp. 466–474, 2019.
- [19] Z. Nazari, M. Nazari, M. S. S. Danish, and D. Kang, "Evaluation of Class Noise Impact on Performance of Machine Learning Algorithms," no. September, 2018.
- [20] Sheena, K. Kumar, and G. Kumar, "Analysis of Feature Selection Techniques: A Data Mining Approach," in *International Conference on Engineering & Technology*, 2016, vol. 4, pp. 17–21.
- [21] P. Gupta and T. Dallas, "Feature Selection and Activity Recognition System Using a Single Triaxial Accelerometer," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1780–1786, Jun. 2014.
- [22] M. Zhang and A. A. Sawchuk, "A Feature Selection-Based Framework for Human Activity Recognition Using Wearable Multimodal Sensors," in *International Conference on Body Area Networks*, 2011, pp. 92–98.
- [23] J.-X. Peng, S. Ferguson, K. Rafferty, and P. D. Kelly, "An efficient feature selection method for mobile devices with application to activity recognition," *Neurocomputing*, vol. 74, no. 17, pp. 3543–3552, Oct. 2011.
- [24] H. Mazaar, E. Emary, and H. Onsi, "Regression-Based Feature Selection on Large Scale Human Activity Recognition," *IJACSA) Int. J. Adv. Comput. Sci.*

- Appl.*, vol. 7, no. 2, pp. 668–674, 2016.
- [25] J. Suto, S. Oniga, and P. P. Sitar, “Comparison of wrapper and filter feature selection algorithms on human activity recognition,” in *2016 6th International Conference on Computers Communications and Control (ICCCC)*, 2016, pp. 124–129.
- [26] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, “A new hybrid filter–wrapper feature selection method for clustering based on ranking,” *Neurocomputing*, vol. 214, pp. 866–880, 2016.
- [27] H. W. Park, D. Li, Y. Piao, and K. H. Ryu, “A Hybrid Feature Selection Method to Classification and Its Application in Hypertension Diagnosis,” no. July, pp. 11–19, 2017.
- [28] R. Panthong and A. Srivihok, “Liver cancer classification model using hybrid feature selection based on class-dependent technique for the central region of Thailand,” *Inf.*, vol. 10, no. 6, 2019.
- [29] K. D. Rajab, “New hybrid features selection method: A case study on websites phishing,” *Secur. Commun. Networks*, vol. 2017, 2017.
- [30] B. M. Abidine, L. Fergani, B. Fergani, and M. Oussalah, “The joint use of sequence features combination and modified weighted SVM for improving daily activity recognition,” *Pattern Anal. Appl.*, vol. 21, no. 1, pp. 119–138, 2018.
- [31] Z. Chen, Q. Zhu, Y. C. Soh, and L. Zhang, “Robust Human Activity Recognition Using Smartphone Sensors via CT-PCA and Online SVM,” *IEEE Trans. Ind. Informatics*, vol. 13, no. 6, pp. 3070–3080, 2017.
- [32] Q. Bouchut, K. Appiah, A. Lotfi, and P. Dickinson, “Ensemble One-vs-One SVM Classifier for Smartphone Accelerometer Activity Recognition,” *Proc. - 20th Int. Conf. High Perform. Comput. Commun. 16th Int. Conf. Smart City 4th Int. Conf. Data Sci. Syst. HPCC/SmartCity/DSS 2018*, pp. 1110–1115, 2019.
- [33] M. Altini, J. Penders, R. Vullers, and O. Amft, “Estimating Energy Expenditure Using Body-Worn Accelerometers: A Comparison of Methods, Sensors Number and Positioning,” *IEEE J. Biomed. Heal. Informatics*, vol. 19, pp. 219–226, 2015.
- [34] F. Palumbo, C. Gallicchio, R. Pucci, and A. Micheli, “Human activity recognition using multisensor data fusion based on Reservoir Computing,” *J.*

- Ambient Intell. Smart Environ.*, vol. 8, no. 2, pp. 87–107, 2016.
- [35] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, “Complex human activity recognition using smartphone and wrist-worn motion sensors,” *Sensors (Switzerland)*, vol. 16, no. 4, pp. 1–24, 2016.
 - [36] D. Castro, W. Coral, C. Rodriguez, J. Cabra, and J. Colorado, “Wearable-based human activity recognition using an IoT Approach,” *J. Sens. Actuator Networks*, vol. 6, no. 4, 2017.
 - [37] C. Chatzaki, M. Pediaditis, G. Vavoulas, and M. Tsiknakis, “Human Daily Activity and Fall Recognition Using a Smartphone’s Acceleration Sensor,” *Commun. Comput. Inf. Sci.*, vol. 736, pp. 100–118, 2017.
 - [38] A. R. Jiménez and F. Seco, “Multi-Event Naive Bayes Classifier for Activity Recognition in the UCAmI Cup †,” in *MDPI proceedings UCAmI 2018*, 2018, pp. 2–7.
 - [39] F. Seco and A. R. Jiménez, “Event-Driven Real-Time Location-Aware Activity Recognition in AAL Scenarios †,” in *MDPI proceedings UCAmI 2018*, 2018, pp. 1–12.
 - [40] Y. Liu, L. Nie, L. Liu, and D. S. Rosenblum, “From action to activity: Sensor-based activity recognition,” *Neurocomputing*, vol. 181, pp. 108–115, 2016.
 - [41] S. A. Kumar, T. Yogesh, M. Prithiv, S. Q. Alam, M. A. B. Hashim, and R. Amutha, “Data Mining Technique based Ambient Assisted Living for Elderly People,” *Proc. 4th Int. Conf. Comput. Methodol. Commun. ICCMC 2020*, no. Iccmc, pp. 505–508, 2020.
 - [42] G. De Leonardis *et al.*, “Human Activity Recognition by Wearable Sensors: Comparison of different classifiers for real-time applications,” *MeMeA 2018 - 2018 IEEE Int. Symp. Med. Meas. Appl. Proc.*, pp. 1–6, 2018.
 - [43] L. H, M.-M. Qmv, B. Qmp, N. Yala, B. Fergani, and A. Fleury, “Towards Improving Feature Extraction and Classification for Activity Recognition on Streaming Data,” *Ambient Intell. Humanised Comput.*, 2016.
 - [44] J. Synnott *et al.*, “Environment Simulation for the Promotion of the Open Data Initiative,” in *2016 IEEE International Conference on Smart Computing, SMARTCOMP 2016*, 2016.
 - [45] O. Kilinc, A. Dalzell, I. Uluturk, and I. Uysal, “Inertial Based Recognition of Daily Activities with ANNs and Spectrotemporal Features,” in *IEEE Machine Learning and Applications (ICMLA)*, 2015, pp. 733–738.

- [46] S. Oniga and S. József, “Optimal Recognition Method of Human Activities Using Artificial Neural Networks,” *Meas. Sci. Rev.*, vol. 15, no. 6, pp. 323–327, 2015.
- [47] S. G. Trost, W.-K. Wong, K. A. Pfeiffer, and Y. Zheng, “Artificial Neural Networks to Predict Activity Type and Energy Expenditure in Youth,” *Med Sci Sport. Exerc.*, vol. 44, no. 9, pp. 1801–1809, 2012.
- [48] M. A. H. Akhand and K. Murase, *Neural Networks Ensembles: Existing Methods and New Techniques*. LAP LAMBERT Academic Publishing, 2010.
- [49] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, pp. 1–18, 2018.
- [50] A. J. C. Sharkey, *Combining Artificial Neural Nets*. Springer, 1999.
- [51] M. P. Sesmero, A. I. Ledezma, and A. Sanchis, “Generating ensembles of heterogeneous classifiers using Stacked Generalization,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 5, no. 1, pp. 21–34, 2015.
- [52] N. Ahmed, J. I. Rafiq, and M. R. Islam, “Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model,” *Sensors (Switzerland)*, vol. 20, no. 1, pp. 1–18, 2020.
- [53] S. Zhang, W. W. Y. Ng, J. Zhang, C. D. Nugent, N. Irvine, and T. Wang, “Evaluation of radial basis function neural network minimizing L-GEM for sensor-based activity recognition,” *J. Ambient Intell. Humaniz. Comput.*, vol. 0, no. 0, p. 0, 2019.
- [54] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, Z. Yu, and S. Member, “Sensor-Based Activity Recognition,” vol. 42, no. 6, pp. 790–808, 2012.
- [55] G. Azkune, A. Almeida, D. López-De-Ipiña, and L. Chen, “Extending Knowledge-Driven Activity Models through Data-Driven Learning Techniques,” *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3115–3128, 2015.
- [56] L. Peng, L. Chen, X. Wu, H. Guo, and G. Chen, “Hierarchical Complex Activity Representation and Recognition Using Topic Model and Classifier Level Fusion,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 6, pp. 1369–1379, 2017.
- [57] A. S. A. Sukor, A. Zakaria, N. A. Rahim, L. M. Kamarudin, R. Setchi, and H. Nishizaki, “A hybrid approach of knowledge-driven and data-driven reasoning for activity recognition in smart homes,” *J. Intell. Fuzzy Syst.*, vol. 36, no. 5, pp. 4177–4188, 2019.

- [58] D. J. Cook and S. K. Das, *Smart Environments: Technology, Protocols, and Applications*. John Wiley & Sons, Inc., 2005.
- [59] B. Ganesan, T. Gowda, A. Al-Jumaily, K. N. K. Fong, S. K. Meena, and R. K. Y. Tong, “Ambient assisted living technologies for older adults with cognitive and physical impairments: A review,” *Eur. Rev. Med. Pharmacol. Sci.*, vol. 23, no. 23, pp. 10470–10481, 2019.
- [60] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, “CASAS: A Smart Home in a Box,” in *IEEE Computing Practices*, 2013, pp. 62–69.
- [61] “The aware home.” [Online]. Available: <http://awarehome.imtc.gatech.edu>.
- [62] S. Helal and C. Chen, “The Gator Tech Smart House: Enabling Technologies and Lessons Learned.”
- [63] “The DOMUS laboratory.” .
- [64] D. Cook *et al.*, “MavHome: An Agent-Based Smart Home,” in *IEEE Pervasive Computing and Communications*, 2003, pp. 521–524.
- [65] C. D. Nugent *et al.*, “Managing Sensor Data in Ambient Assisted Living,” *J. Comput. Sci. Eng.*, vol. 5, no. 3, pp. 237–245, 2011.
- [66] J. Synnott, C. Nugent, and P. Jeffers, “Simulation of Smart Home Activity Datasets,” *Sensors (Switzerland)*, vol. 15, no. 6, pp. 14162–14179, 2015.
- [67] P. Rashidi and A. Mihailidis, “A Survey on Ambient Assisted Living Tools for Older Adults,” *IEEE J. Biomed. Heal. Informatics*, vol. 17, no. 3, pp. 579–590, 2013.
- [68] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, “The Gator Tech Smart House: a programmable pervasive space,” *IEEE Comput.*, vol. 38, no. 3, pp. 50–60, 2005.
- [69] P. Chahuara, “Using MLN for Activity Recognition,” in *Ambient Intelligence: Third International Joint Conference, AmI*, 2012, pp. 185–188.
- [70] A. Bulling, U. Blanke, and B. Schiele, “A Tutorial on Human Activity Recognition using Body-Worn Inertial Sensors,” *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, 2014.
- [71] J. Morales and D. Akopian, “Physical activity recognition by smartphones, a survey,” *Biocybern. Biomed. Eng.*, vol. 37, no. 3, pp. 388–400, Jan. 2017.
- [72] D. R. MB *et al.*, “A Comparison of Activity Classification in Younger and Older Cohorts using a Smartphone,” *Physiol. Meas.*, vol. 35, no. 11, pp. 2269–86, 2014.

- [73] V. Q. Viet, G. Lee, and D. Choi, “Fall Detection Based on Movement and Smart Phone Technology,” in *IEEE Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, 2012, pp. 1–4.
- [74] D. J. Cook and N. C. Krishnan, *Activity Learning: Discovering, Recognizing and Predicting Human Behavior from Sensor Data*, 1st ed. Wiley, 2015.
- [75] S. Khaled, F. Attal, M. Samer, M. Khalil, and A. Yacine, “Physical Activity Recognition Using Inertial Wearable Sensors - A Review of Supervised Classification Algorithms,” in *International Conference on Advances in Biomedical Engineering (ICABME)*, 2015, pp. 313–316.
- [76] M. Gochoo, T. Tan, and S. Huang, “DCNN-Based Elderly Activity Recognition Using Binary Sensors,” in *International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2017.
- [77] D. Singh, E. Merdivan, and S. Hanke, “Convolutional and Recurrent Neural Networks for Activity Recognition in Smart Environment,” in *Towards Integrative Machine Learning and Knowledge Extraction*, 2017, pp. 194–209.
- [78] A. Wang, G. Chen, and C. Shang, “Human Activity Recognition in a Smart Home Environment with Stacked Denoising Autoencoders,” in *Web-Age Information Management (WAIM)*, 2016, pp. 29–40.
- [79] M. Espinilla, J. Medina, and C. Nugent, “UCAmI Cup. Analyzing the UJA Human Activity Recognition Dataset of Activities of Daily Living †,” in *MDPI proceedings UCAmI 2018*, 2018, pp. 1–14.
- [80] Y. Roh, G. Heo, and S. E. Whang, “A Survey on Data Collection for Machine Learning: a Big Data - AI Integration Perspective,” pp. 1–19, 2018.
- [81] K. Connelly *et al.*, “The Future of Pervasive Health,” *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 16–20, 2017.
- [82] C. Pelletier, S. Valero, J. Inglada, N. Champion, C. M. Sicre, and G. Dedieu, “Effect of training class label noise on classification performances for land cover mapping with satellite image time series,” *Remote Sens.*, vol. 9, no. 2, 2017.
- [83] F. Cruciani, I. Cleland, C. Nugent, P. McCullagh, K. Synnes, and J. Hallberg, “Automatic annotation for human activity recognition in free living using a smartphone,” *Sensors (Switzerland)*, vol. 18, no. 7, pp. 1–20, 2018.
- [84] R. Indika and P. Wickramasinghe, “Attribute Noise, Classification Technique,

- and Classification Accuracy,” in *Data Analytics and Decision Support for Cybersecurity*, Springer, Cham, 2017, pp. 201–220.
- [85] B. Quigley, M. Donnelly, G. Moore, and L. Galway, “A Comparative Analysis of Windowing Approaches in Dense Sensing Environments,” *Proceedings*, vol. 2, no. 19, p. 1245, 2018.
- [86] N. A. Capela, E. D. Lemaire, and N. Baddour, “Feature Selection for Wearable Smartphone-Based Human Activity Recognition with Able bodied, Elderly, and Stroke Patients,” *PLoS One*, vol. 10, no. 4, p. e0124414, Apr. 2015.
- [87] Y. Xu *et al.*, “Learning multi-level features for sensor-based human action recognition,” *Pervasive Mob. Comput.*, vol. 40, pp. 324–338, 2017.
- [88] D. Roggen, A. Calatroni, M. Rossi, and T. Holleczeck, “Collecting complex activity data sets in highly rich networked sensor environments,” in *Networked Sensing Systems (INSS’10)*, 2010.
- [89] P. Zappi, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, “Activity recognition from on-body sensors by classifier fusion: Sensor scalability and robustness,” *Proc. 2007 Int. Conf. Intell. Sensors, Sens. Networks Inf. Process. ISSNIP*, pp. 281–286, 2007.
- [90] J. Kwapisz, G. Weiss, and S. Moore, “Activity Recognition using Cell Phone Accelerometers,” in *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data (at KDD-10)*, 2010.
- [91] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Comput. Electr. Eng.*, vol. 40, pp. 16–28, 2014.
- [92] D. Oreski and T. Novosel, “Comparison of Feature Selection Techniques in Knowledge Discovery Process,” *TEM J.*, vol. 3, no. 4, pp. 285–290, 2014.
- [93] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, “Benchmark for filter methods for feature selection in high-dimensional classification data,” *Comput. Stat. Data Anal.*, vol. 143, 2020.
- [94] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, “Relief-based feature selection: Introduction and review,” *J. Biomed. Inform.*, vol. 85, no. July, pp. 189–203, 2018.
- [95] R. Panthong and A. Srivihok, “Wrapper Feature Subset Selection for Dimension Reduction Based on Ensemble Learning Algorithm,” in *Procedia Computer Science. Special Issue: The Third Information Systems International Conference*, 2015, vol. 72, pp. 162–169.

- [96] M. M. Sakr, M. A. Tawfeeq, and A. B. El-Sisi, "Filter Versus Wrapper Feature Selection for Network Intrusion Detection System," *Proc. - 2019 IEEE 9th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2019*, pp. 209–214, 2019.
- [97] X. Wu *et al.*, *Top 10 algorithms in data mining*, vol. 14, no. 1. 2008.
- [98] S. Shaily and V. Mangat, "The Hidden Markov Model and its Application to Human Activity Recognition," in *IEEE International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*, 2015, pp. 1–4.
- [99] G. Mountrakis, J. Im, and C. Ogole, "Support Vector Machines in Remote Sensing: A Review," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 3, pp. 247–259, 2011.
- [100] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Ambient Assisted Living and Home Care*, 2012, pp. 216–223.
- [101] M. Kumari and S. Soni, "A Review of classification in Web Usage Mining using K- Nearest Neighbour," *Adv. Comput. Sci. Technol.*, vol. 10, no. 5, pp. 1405–1416, 2017.
- [102] H. A. Abu Alfeilat *et al.*, "Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review," *Big Data*, vol. 7, no. 4, pp. 221–248, 2019.
- [103] M. N. Shah Zainudin, M. N. Sulaiman, N. Mustapha, and T. Perumal, "One-against-all binarization classification strategy to recognize interclass similarities activities from several sensor positions," *J. Eng. Sci. Technol.*, vol. 13, no. 8, pp. 2549–2568, 2018.
- [104] M. S. Ahmed, M. Shahjaman, M. M. Rana, and M. N. H. Mollah, "Robustification of Naïve Bayes Classifier and Its Application for Microarray Gene Expression Data Analysis," *Biomed Res. Int.*, vol. 2017, 2017.
- [105] G. Chetty and M. White, "Body sensor networks for human activity recognition," *3rd Int. Conf. Signal Process. Integr. Networks, SPIN 2016*, pp. 660–665, 2016.
- [106] P. Asghari, E. Soleimani, and E. Nazerfard, "Online human activity recognition employing hierarchical hidden Markov models," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 3, pp. 1141–1152, 2020.
- [107] S. Kamal, A. Jalal, and D. Kim, "Depth images-based human detection,

- tracking and activity recognition using spatiotemporal features and modified HMM,” *J. Electr. Eng. Technol.*, vol. 11, no. 6, pp. 1857–1862, 2016.
- [108] P. LAGO and S. INOUE, “A Hybrid Model Using Hidden Markov Chain and Logic Model for Daily Living Activity Recognition,” in *Proceedings*, 2018, vol. 2, no. 19, p. 1266.
- [109] F. Al-shargie, T. B. Tang, N. Badruddin, and M. Kiguchi, “Towards multilevel mental stress assessment using SVM with ECOC: an EEG approach,” *Med. Biol. Eng. Comput.*, vol. 56, no. 1, pp. 125–136, 2018.
- [110] Y. Y. Song and Y. Lu, “Decision tree methods: applications for classification and prediction,” *Shanghai Arch. Psychiatry*, vol. 27, no. 2, pp. 130–135, 2015.
- [111] S. Böttcher, P. M. Scholl, and K. Van Laerhoven, “Detecting Transitions in Manual Tasks from Wearables: An Unsupervised Labeling Approach,” *Informatics*, vol. 5, no. 16, 2018.
- [112] O. Terzo and L. Barolli, “Complex, Intelligent, and Software Intensive Systems,” in *11th International Conference on Complex, Intelligent, and Software Intensive Systems*, 2017, pp. 25–27.
- [113] D. S. Yeung, J. C. Li, W. W. Y. Ng, and P. P. K. Chan, “MLPNN Training via a Multiobjective Optimization of Training Error and Stochastic Sensitivity,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 27, no. 5, pp. 978–992, 2016.
- [114] B. T. Pham, M. D. Nguyen, K. T. T. Bui, I. Prakash, K. Chapi, and D. T. Bui, “A novel artificial intelligence approach based on Multi-layer Perceptron Neural Network and Biogeography-based Optimization for predicting coefficient of consolidation of soil,” *Catena*, vol. 173, no. July, pp. 302–311, 2019.
- [115] W. W. Myo, W. Wettayaprasit, and P. Aiyarak, “Designing classifier for human activity recognition using artificial neural network,” *2019 IEEE 4th Int. Conf. Comput. Commun. Syst. ICCCS 2019*, pp. 81–85, 2019.
- [116] S. Greengard, “GPUs Reshape Computing,” *Communications of the ACM*, vol. 59, no. 9, pp. 14–16, 2016.
- [117] I. Sutskever, O. Vinyals, and L. Q. V, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.
- [118] M. Wu and L. Chen, “Image recognition based on deep learning,” *Proc. - 2015 Chinese Autom. Congr. CAC 2015*, pp. 542–546, 2016.

- [119] Y. Dong and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*, 1st ed. Springer, London, 2016.
- [120] J. Suto and S. Oniga, “Efficiency investigation of artificial neural networks in human activity recognition,” *J. Ambient Intell. Humaniz. Comput.*, vol. 0, no. 0, p. 0, 2017.
- [121] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep Learning for Sensor-based Activity Recognition: A Survey,” *Pattern Recognit. Lett.*, 2017.
- [122] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and K. Shonali, “Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition,” in *IJCAI’15 Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, no. Ijcai, pp. 3995–4001.
- [123] F. J. Ordóñez and D. Roggen, “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition,” *Sensors (Switzerland)*, vol. 16, no. 1, 2016.
- [124] M. Zeng *et al.*, “Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors,” in *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, 2014, pp. 197–205.
- [125] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [126] N. Y. Hammerla, S. Halloran, and T. Plötz, “Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables.”
- [127] T. Shu, S. Todorovic, and S.-C. Zhu, “CERN: Confidence-Energy Recurrent Network for Group Activity Recognition,” 2017.
- [128] J. Suto and S. Oniga, “Efficiency investigation from shallow to deep neural network techniques in human activity recognition,” *Cogn. Syst. Res.*, vol. 54, pp. 37–49, 2019.
- [129] N. Rooney, D. Patterson, and C. Nugent, “Ensemble Learning for Regression,” *Encycl. Data Warehous. Mining, Second Ed.*, vol. II, pp. 777–782, 2010.
- [130] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. De Sousa, “Ensemble Approaches for Regression: A Survey,” *ACM Comput. Surv.*, vol. 45, no. 1, pp. 1–40, 2012.
- [131] Z. Feng, L. Mo, and M. Li, “A Random Forest-based ensemble method for activity recognition,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*,

- vol. 2015-Novem, pp. 5074–5077, 2015.
- [132] Y. J. Kim, B. N. Kang, and D. Kim, “Hidden Markov Model Ensemble for Activity Recognition Using Tri-Axis Accelerometer,” *Proc. - 2015 IEEE Int. Conf. Syst. Man, Cybern. SMC 2015*, pp. 3036–3041, 2016.
- [133] H. Sagha, H. Bayati, J. D. R. Millán, and R. Chavarriaga, “On-line anomaly detection and resilience in classifier ensembles,” *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 1916–1927, 2013.
- [134] E. P. Ijjina and C. Krishna Mohan, “Hybrid deep neural network model for human action recognition,” *Appl. Soft Comput. J.*, 2016.
- [135] I. Hwang, H. M. Park, and J. H. Chang, “Ensemble of deep neural networks using acoustic environment classification for statistical model-based voice activity detection,” *Comput. Speech Lang.*, 2016.
- [136] Y. Guan and T. Ploetz, “Ensembles of Deep LSTM Learners for Activity Recognition using Wearables,” *ACM IMMUT*, vol. 0, no. 0, 2017.
- [137] H. F. Nweke, Y. W. Teh, G. Mujtaba, and M. A. Al-garadi, “Data fusion and multiple classifier systems for human activity detection and health monitoring,” *Inf. Fusion*, vol. 46, pp. 147–170, 2019.
- [138] M. Wozniak, W. Wozniak, M. Graña, and E. Corchado, “A survey of multiple classifier systems as hybrid systems,” 2014.
- [139] J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, and L. I. Kuncheva, “Random Balance: Ensembles of variable priors classifiers for imbalanced data,” *Knowledge-Based Syst.*, vol. 85, pp. 96–111, 2015.
- [140] W. Feng, W. Huang, and J. Ren, “Class imbalance ensemble learning based on the margin theory,” *Appl. Sci.*, vol. 8, no. 5, 2018.
- [141] S. Ahmed, A. Mahbub, F. Rayhan, R. Jani, S. Shatabda, and D. M. Farid, “Hybrid Methods for Class Imbalance Learning Employing Bagging with Sampling Techniques,” *2nd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS 2017*, pp. 1–5, 2018.
- [142] K. N. K. A. Rahim, I. Elamvazuthi, L. I. Izhar, and G. Capi, “Classification of human daily activities using ensemble methods based on smartphone inertial sensors,” *Sensors (Switzerland)*, vol. 18, no. 12, 2018.
- [143] Y. W. Xue, J. Liu, J. Chen, Y. T. Zhang, and R. Cao, “Feature Grouping Based on Ga and L-Gem for Human Activity Recognition,” *Proc. - Int. Conf. Mach. Learn. Cybern.*, vol. 1, pp. 44–49, 2018.

- [144] M. Farooq and E. Sazonov, "Detection of chewing from piezoelectric film sensor signals using ensemble classifiers," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 4929–4932.
- [145] E. Mohammadi, Q. M. Jonathan Wu, and M. Saif, "Human activity recognition using an ensemble of support vector machines," in *2016 International Conference on High Performance Computing & Simulation (HPCS)*, 2016, pp. 549–554.
- [146] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Comput. Sci.*, vol. 34, no. C, pp. 450–457, 2014.
- [147] E. Vogiatzaki and A. Krukowski, *Modern stroke rehabilitation through e-health-based entertainment*. 2015.
- [148] T. Lv, X. Wang, L. Jin, Y. Xiao, and M. Song, "Margin-based deep learning networks for human activity recognition," *Sensors (Switzerland)*, vol. 20, no. 7, 2020.
- [149] A. Reiss and D. Stricker, "Introducing a New Benchmarked Dataset for Activity Monitoring," in *The 16th IEEE International Symposium on Wearable Computers (ISWC)*, 2012.
- [150] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones," *Appl. Sci.*, vol. 7, no. 10, 2017.
- [151] C. A. Ronao and S.-B. Cho, "Human Activity Recognition with Smartphone Sensors using Deep Learning Neural Networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, 2016.
- [152] A. Gosain and S. Sardana, "Handling class imbalance problem using oversampling techniques: A review," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 79–85.
- [153] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, 2017.
- [154] N. Ofek, L. Rokach, R. Stern, and A. Shabtai, "Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem,"

- Neurocomputing*, vol. 243, pp. 88–102, 2017.
- [155] W. C. Lin, C. F. Tsai, Y. H. Hu, and J. S. Jhang, “Clustering-based undersampling in class-imbalanced data,” *Inf. Sci. (Ny)*, vol. 409–410, pp. 17–26, 2017.
- [156] J. Zhang, W. W. Y. Ng, S. Zhang, and C. D. Nugent, “Two-Phase Resampling for Noisy Imbalanced Multi-class Classification Problems and its Application in Human Activity Recognition,” *IEEE Trans. Neural Networks Learn. Syst. Spec. Issue Recent Adv. Theory, Methodol. Appl. Imbalanced Learn.*, vol. 14, no. 8, pp. 1–13, 2015.
- [157] N. A. Sakr, M. Abu-Elkheir, A. Atwan, and H. H. Soliman, “Data driven recognition of interleaved and concurrent human activities with nonlinear characteristics,” *J. Intell. Fuzzy Syst.*, vol. 37, no. 4, pp. 5573–5588, 2019.
- [158] M. Rawashdeh, M. G. Al Zamil, S. Samarah, M. S. Hossain, and G. Muhammad, “A knowledge-driven approach for activity recognition in smart homes based on activity profiling,” *Futur. Gener. Comput. Syst.*, vol. 107, pp. 924–941, 2020.
- [159] V. Sessions and M. Valtorta, “The Effects of Data Quality on Machine Learning Algorithms,” in *Proceedings of the 2006 International Conference on Information Quality, ICIQ 2006*, 2006.
- [160] A. A. Folleco, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “Identifying Learners Robust to Low Quality Data,” *Informatica*, vol. 33, pp. 245–259, 2009.
- [161] I. Cleland, “Data Preparation.” pp. 1–200, 2018.
- [162] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*, 1st ed. Springer, 2005.
- [163] F. Gorunescu, *Data Mining: Concepts, Models and Techniques*, 1st ed. Springer, 2011.
- [164] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Elsevier Inc., 2012.
- [165] C. C. Aggarwal, “An Introduction to Outlier Analysis,” in *Outlier Analysis*, Springer International Publishing AG, 2017, pp. 1–34.
- [166] H. Ghallab, H. Fahmy, and M. Nasr, “Detection outliers on internet of things using big data technology,” *Egypt. Informatics J.*, no. xxxx, pp. 1–8, 2019.
- [167] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, “Detecting outliers: Do

- not use standard deviation around the mean, use absolute deviation around the median,” *J. Exp. Soc. Psychol.*, vol. 49, no. 4, pp. 764–766, 2013.
- [168] Z. Niu, S. Shi, J. Sun, and X. He, “A Survey of Outlier Detection Methodologies and Their Applications,” in *Artificial Intelligence and Comp. Intelligence*, 2011, pp. 380–387.
- [169] S. Sun *et al.*, “Optimization of support vector regression model based on outlier detection methods for predicting electricity consumption of a public building WSHP system,” *Energy Build.*, vol. 151, pp. 35–44, 2017.
- [170] M. Munoz-Organero, “Outlier Detection in Wearable Sensor Data for Human Activity Recognition (HAR) Based on DRNNs,” *IEEE Access*, vol. 7, pp. 74422–74436, 2019.
- [171] P. Pirzada, N. White, and A. Wilde, “Sensors in smart homes for independent living of the elderly,” *5th Int. Multi-Topic ICT Conf. Technol. Futur. Gener. IMTIC 2018 - Proc.*, pp. 1–8, 2018.
- [172] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, “Window Size Impact in Human Activity Recognition,” *Sensors (Basel)*, vol. 14, no. 4, pp. 6474–6499, Apr. 2014.
- [173] M. Hoogendoorn and B. Funk, “Machine Learning for the Quantified Self: On the Art of Learning from Sensory Data,” in -, 2018, pp. 64–67.
- [174] A. . Khan, Y. . Lee, and S. . Lee, “Accelerometer’s Position Free Human Activity Recognition Using a Hierarchical Recognition Model,” in *IEEE HealthCom*, 2010.
- [175] A. Mannini, M. Rosenberger, W. L. Haskell, A. M. Sabatini, and S. S. Intille, “Activity Recognition in Youth Using Single Accelerometer Placed at Wrist or Ankle,” *Med. Sci. Sport. Exerc.*, vol. 49, no. 4, pp. 801–812, 2017.
- [176] N. Settouti, M. E. A. Bechar, and M. A. Chikh, “Statistical Comparisons of the Top 10 Algorithms in Data Mining for Classification Task,” *Int. J. Interact. Multimed. Artif. Intell.*, vol. 4, no. 1, p. 46, 2016.
- [177] Y. Xia, C. Liu, B. Da, and F. Xie, “A novel heterogeneous ensemble credit scoring model based on bstacking approach,” *Expert Syst. Appl.*, vol. 93, pp. 182–199, 2018.
- [178] Z. Wu, Q. Xu, J. Li, C. Fu, Q. Xuan, and Y. Xiang, “Passive Indoor Localization Based on CSI and Naive Bayes Classification,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 48, no. 9, pp. 1566–1577, 2018.

- [179] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, “A study of the effect of different types of noise on the precision of supervised learning techniques,” *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 275–306, 2010.
- [180] J. Hu and P. Wang, “Noise robustness analysis of performance for EEG-based driver fatigue detection using different entropy feature sets,” *Entropy*, vol. 19, no. 8, 2017.
- [181] A. Altaher, “Phishing Websites Classification using Hybrid SVM and KNN Approach,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 90–95, 2017.
- [182] A. Das Antar, M. Ahmed, and M. A. R. Ahad, “Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: A review,” *2019 Jt. 8th Int. Conf. Informatics, Electron. Vision, ICIEV 2019 3rd Int. Conf. Imaging, Vis. Pattern Recognition, icIVPR 2019 with Int. Conf. Act. Behav. Comput. ABC 2019*, pp. 134–139, 2019.
- [183] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, “Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges and Opportunities,” *Assoc. Comput. Mach.*, vol. 37, no. 4, 2018.
- [184] Q. Ni *et al.*, “Leveraging Wearable Sensors for Human Daily Activity Recognition with Stacked Denoising Autoencoders,” *Sensors*, vol. 20, pp. 1–22, 2020.
- [185] “1st UCAmI Cup - Analysing the UJAEN Human Activity Recognition Dataset,” 2018. [Online]. Available: <http://mamilab.esi.uclm.es/ucami2018/UCAmICup.html>. [Accessed: 22-Oct-2020].
- [186] N. Karvonen and D. Kleyko, “A Domain Knowledge-Based Solution for Human Activity Recognition: The UJA Dataset Analysis †,” in *MDPI proceedings UCAmI 2018*, 2018, pp. 1–8.
- [187] J. D. Cerón, D. M. López, and B. M. Eskofier, “Human Activity Recognition Using Binary Sensors , BLE Beacons , an Intelligent Floor and Acceleration Data : A Machine Learning Approach †,” in *MDPI proceedings UCAmI 2018*, 2018, pp. 1–7.
- [188] M. A. Razzaq, I. Cleland, C. Nugent, and S. Lee, “Multimodal Sensor Data Fusion for Activity Recognition Using Filtered Classifier †,” in *MDPI proceedings UCAmI 2018*, 2018, pp. 1–11.
- [189] D. Ding, R. A. Cooper, P. F. Pasquina, and L. Fici-Pasquina, “Sensor

- technology for smart homes,” *Maturitas*, vol. 69, no. 2, pp. 131–136, 2011.
- [190] M. Amiribesheli, A. Benmansour, and A. Bouchachia, “A review of smart homes in healthcare,” *J. Ambient Intell. Humaniz. Comput.*, vol. 6, no. 4, pp. 495–517, 2015.
- [191] I. Witten, E. Frank, and M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Elsevier, 2011.
- [192] Y. Zhang, X. Ren, and J. Zhang, “Intrusion detection method based on information gain and ReliefF feature selection,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2019-July, no. July, pp. 1–5, 2019.
- [193] A. Katrutsa and V. Strijov, “Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria,” *Expert Syst. Appl.*, vol. 76, pp. 1–11, 2017.
- [194] B. Krawczyk and M. Woźniak, “Untrained weighted classifier combination with embedded ensemble pruning,” *Neurocomputing*, vol. 196, pp. 14–22, 2016.
- [195] C. Ranjan, “Rules-of-thumb for building a Neural Network,” 2019. .
- [196] M. Mohandes, M. Deriche, and S. O. Aliyu, “Classifiers Combination Techniques: A Comprehensive Review,” *IEEE Access*, vol. 6, pp. 19626–19639, 2018.
- [197] L. Yijing, G. Haixiang, L. Xiao, L. Yanan, and L. Jinling, “Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data,” *Knowledge-Based Syst.*, vol. 94, pp. 88–104, 2016.
- [198] A. Petrakova, M. Affenzeller, and G. Merkurjeva, “Heterogeneous versus Homogeneous Machine Learning Ensembles,” *Inf. Technol. Manag. Sci.*, vol. 18, no. 1, pp. 135–140, 2016.
- [199] Z. Lu, X. Wu, and J. C. Bongard, “Active learning through adaptive heterogeneous ensembling,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 368–381, 2015.
- [200] Q. Ni, L. Zhang, and L. Li, “A Heterogeneous Ensemble Approach for Activity Recognition with Integration of Change Point-Based Data Segmentation,” *Appl. Sci.*, vol. 8, no. 9, p. 1695, 2018.
- [201] I. Fatima, M. Fahim, Y.-K. Lee, and S. Lee, “Classifier ensemble optimization for human activity recognition in smart homes,” in *Proceedings of the 7th*

- International Conference on Ubiquitous Information Management and Communication - ICUIMC '13*, 2013, vol. 7, no. 11, pp. 1–7.
- [202] V. L. Diengdoh, S. Ondeï, M. Hunt, and B. W. Brook, “A validated ensemble method for multinomial land-cover classification,” *Ecol. Inform.*, vol. 56, no. October 2019, 2020.
- [203] R. Granada, J. Monteiro, R. Barros, and F. Meneguzzi, “A Deep Neural Architecture for Kitchen Activity Recognition,” in *International Artificial Intelligence Research Society Conference*, 2017, pp. 56–61.
- [204] W. H. Chen and Y. Chen, “An ensemble approach to activity recognition based on binary sensor readings,” *2017 IEEE 19th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2017*, vol. 2017-Decem, pp. 1–5, 2017.
- [205] N. Irvine, C. Nugent, S. Zhang, H. U. I. Wang, W. W. Ng, and I. A. N. C. Espinilla, “The Impact of Dataset Quality on the Performance of Data-Driven Approaches for Human Activity Recognition,” in *FLINS on Data science and knowledge engineering for sensing decision support*, 2018, pp. 1–9.
- [206] N. Irvine, C. Nugent, S. Zhang, H. Wang, and W. W. Y. Ng, “Neural network ensembles for sensor-based human activity recognition within smart environments,” *Sensors (Switzerland)*, vol. 20, no. 1, 2020.

Appendix 1



Pervasive Computing in Healthcare

Data Collection Protocol

Data collection protocol

This document summarises the data collection protocol you must follow when collecting the experimental data for assignment 2.

You will have already been assigned to one of the following scenario groups. Please ensure that you collect data for the activities you have been assigned. We have produced a video to show how the shimmer should be calibrated and configured and how the data should be collected for each of the activities:

https://youtu.be/e_WN_hlh_xo

Please read the following instructions carefully.

Data collection

Data collection can be split into 2 components; Calibration and Scenario activities. The first component is a verification of the calibration procedure all groups must collect this data. You will then collect data during three different activities depending on the group below.

Collection of Calibration Data

The purpose of the calibration data is to allow you to assess whether the shimmer has been correctly calibrated or not. **Following appropriate calibration** of the Shimmer, you should collect one continuous file of data. During this time, you should place the shimmer on a flat surface, and leave it untouched for **5 seconds**. You should orientate the shimmer in each of the **6 orientations** below. Once plotted, the data should look something like that plotted in Fig 1. Save the file as **B001234567_Calibration.csv**.

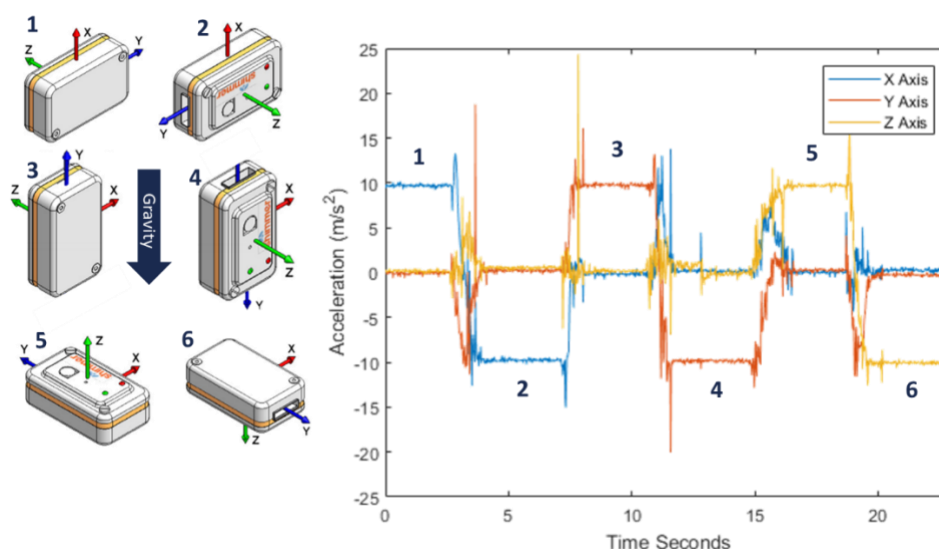


Fig 1. Expected output of calibrated accelerometer in each orientation.

Collection of Scenarios

1. Self-care Scenario (hair grooming, washing hands and teeth brushing)
2. Exercise Scenario (cardio) (walking, jogging, stepping)

3. House cleaning Scenario (ironing, window washing and dish washing)
4. Exercise Scenario (weights) (arm curls, deadlift and lateral arm raise)
5. Sport Scenario (pass, bounce, catch)
6. Food Preparation Scenario (mixing food in a bowl, chopping vegetables, sieving flour)

Methodology

Data for all scenarios will be collected using the same methodology. Only the activities will be different.

1. Calibrate the shimmer as outlined in the week 2 practical and the video. Double check the calibration to ensure it is correct.
2. Data will be collected using a single shimmer placed on the **Dominant** wrist. The orientation will be fixed using strapping/ elastic bands. The shimmer should be fixed to the wrist with the shimmer logo facing upward and inwards as shown by the images below (Fig2).

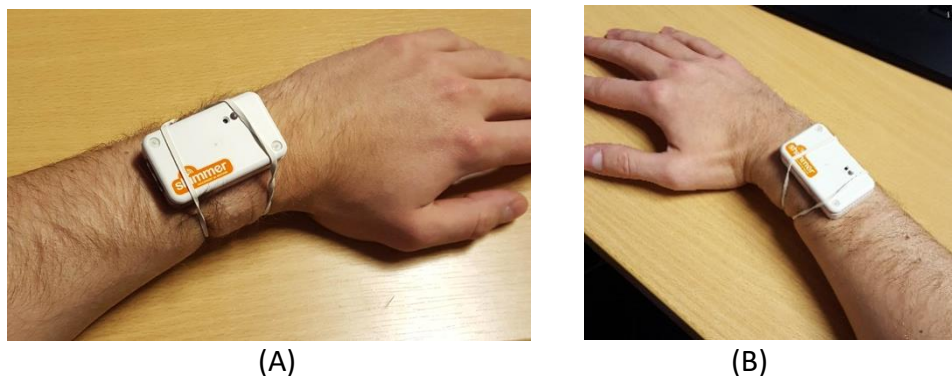


Fig 1. Image shown the Shimmer attached to a) the Left and b) the Right wrist. In both cases the shimmer logo should face upwards and inwards.

3. Data will be collected through Shimmer Connect. The shimmer should be configured in the following manner. This is demonstrated in Fig. 3.
 - a. Accelerometer should be the **only** sensor recorded
 - b. A sampling rate of **51.2Hz** should be used
 - c. The Range should be set to **+6g**
 - d. Logging Delimiter format: **Comma**

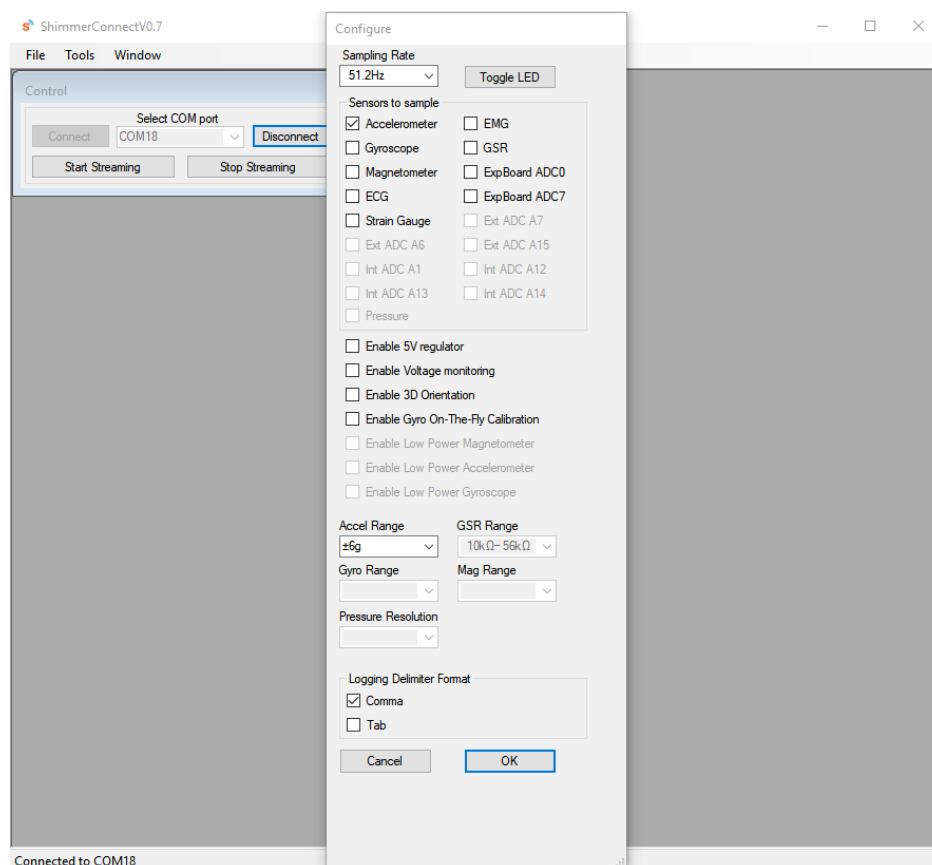


Fig. 3. Screenshot of Shimmer Connect showing the Configuration screen. Note only the Accelerometer sensor is selected. The Sampling Rate is 51.2Hz. The Range is set to +6g. The Logging Delimiter is Comma.

4. The shimmer is now configured to collect data. Before starting, ensure you have selected to save the data to file (Tools -> Save to CSV) and that you are saving the file in a safe place.

The File name for the CSV should follow the following format:

B001234567_X_ActivityName.csv

Where B001234567 Should be your Student B number. X should be the Scenario you have been assigned to (i.e. 1 for Self care) and the ActivityName should be the name of the activity you are recoding in that file (i.e. HairGrooming)

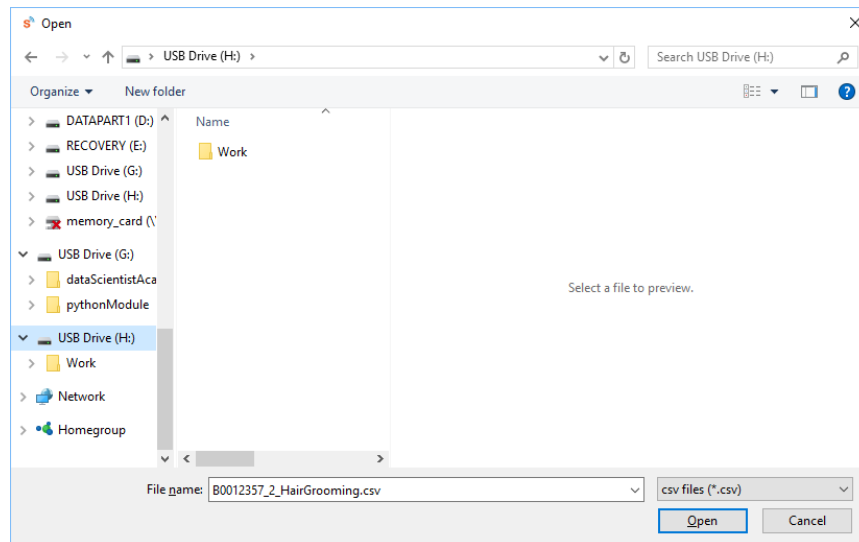


Fig.4. Check Save to CSV prior to streaming the data. Save the file with the appropriate file name **B001234567_X_ActivityName.csv**

5. We can now begin to collect data for the first activity. Click Start Streaming and begin undertaking the First activity in the manner described previously in the videos. Each activity should be recorded for **2 minutes** each. We recommend you time this on your phone. **Once you have finished collecting the data for the first activity click Stop Streaming.**

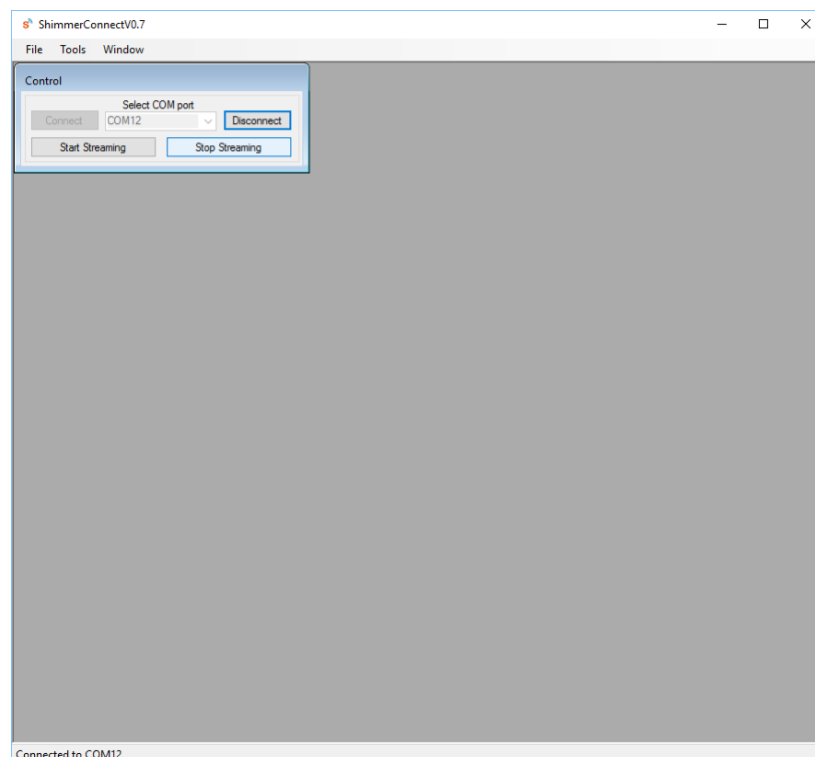
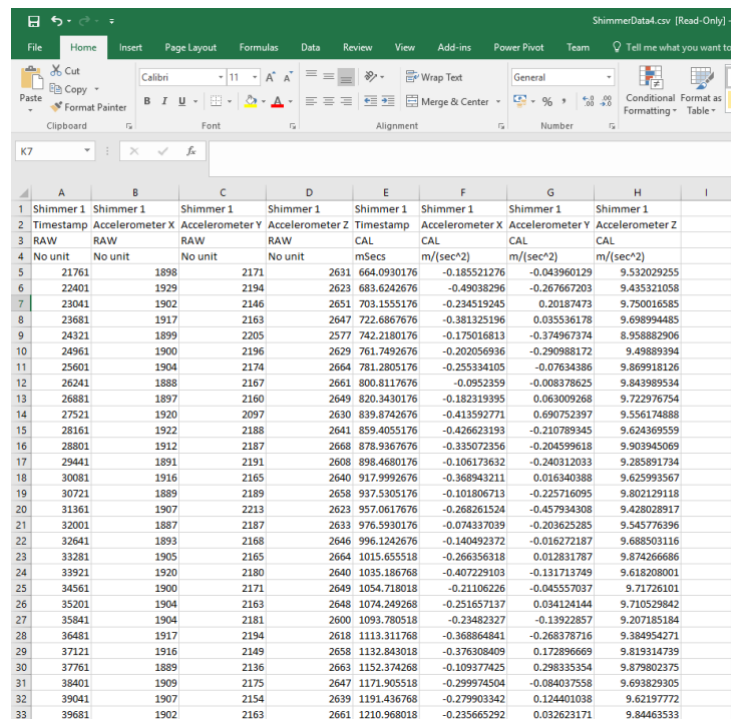


Fig.5. Stop streaming between each activity.

Navigate to where the File is saved. Open the file in excel and ensure that the Format matches the one below. Note, Only the Raw and Calibrated Accelerometer data is collected. If the format of your file does not match this, check the configuration settings.



	Shimmer 1	Shimmer 1	Shimmer 1	Shimmer 1	Shimmer 1	Shimmer 1	Shimmer 1	Shimmer 1
	Accelerometer X	Accelerometer Y	Accelerometer Z	Timestamp	Accelerometer X	Accelerometer Y	Accelerometer Z	
	RAW	RAW	RAW	RAW	CAL	CAL	CAL	CAL
	No unit	No unit	No unit	No unit	mSecs	m/(sec^2)	m/(sec^2)	m/(sec^2)
5	21761	1898	2171	2631	664.0930176	-0.185521276	-0.043960129	9.532029255
6	22401	1929	2194	2623	683.6242676	-0.49038296	-0.267667203	9.435321058
7	23041	1902	2146	2651	703.1555176	-0.234519245	0.20187473	9.750016585
8	23681	1917	2163	2647	722.6867676	-0.381325196	0.035536178	9.698994485
9	24321	1899	2205	2577	742.2180176	-0.175016813	-0.374967374	8.958882906
10	24961	1900	2196	2629	761.7492676	-0.202056936	-0.290988172	9.49889394
11	25601	1904	2174	2664	781.2805176	-0.255334105	-0.07634386	9.869918126
12	26241	1888	2167	2661	800.8117676	-0.0952359	-0.008378625	9.84398534
13	26881	1897	2160	2649	820.3430176	-0.182319395	0.063009268	9.722976754
14	27521	1920	2097	2630	839.8742676	-0.413592771	0.690752397	9.556174888
15	28161	1922	2188	2641	859.4055176	-0.426623193	-0.210789345	9.624369559
16	28801	1912	2187	2668	878.9367676	-0.335072356	-0.204599618	9.903945069
17	29441	1891	2191	2608	898.4680176	-0.106173632	-0.240312033	9.285891734
18	30081	1916	2165	2640	917.9992676	-0.368943211	0.016340388	9.625993567
19	30721	1889	2189	2658	937.5305176	-0.101806713	-0.225716095	9.802129118
20	31361	1907	2213	2623	957.0617676	-0.268261524	-0.457934308	9.428028917
21	32001	1887	2187	2633	976.5930176	-0.074337039	-0.203625285	9.545776396
22	32641	1893	2168	2646	996.1242676	-0.140492372	-0.016272187	9.688503116
23	33281	1905	2165	2664	1015.655518	-0.266356318	0.012831787	9.874266686
24	33921	1920	2180	2640	1035.186768	-0.407229103	-0.131713749	9.618208001
25	34561	1900	2171	2649	1054.718018	-0.21106226	-0.045557037	9.71726101
26	35201	1904	2163	2648	1074.249268	-0.251657137	0.034124144	9.710529842
27	35841	1904	2181	2600	1093.780518	-0.23482327	-0.13922857	9.207185184
28	36481	1917	2194	2618	1113.311768	-0.368864841	-0.268378716	9.384954271
29	37121	1916	2149	2658	1132.843018	-0.376308409	0.172896669	9.819314739
30	37761	1889	2136	2663	1152.374268	-0.109377425	0.298335354	9.879802375
31	38401	1909	2175	2647	1171.905518	-0.299974504	-0.084037558	9.693829305
32	39041	1907	2154	2639	1191.436768	-0.279903942	0.124401038	9.62197772
33	39681	1902	2163	2661	1210.968018	-0.235665292	0.052623171	9.84463533

Fig.6. Screen shot of the data in Excel. Note only the Raw and Calibrated Accel data is recorded.

- Before collecting data for the next activity, uncheck and recheck Save to CSV. Ensure you provide a new Filename with the same structure as before.

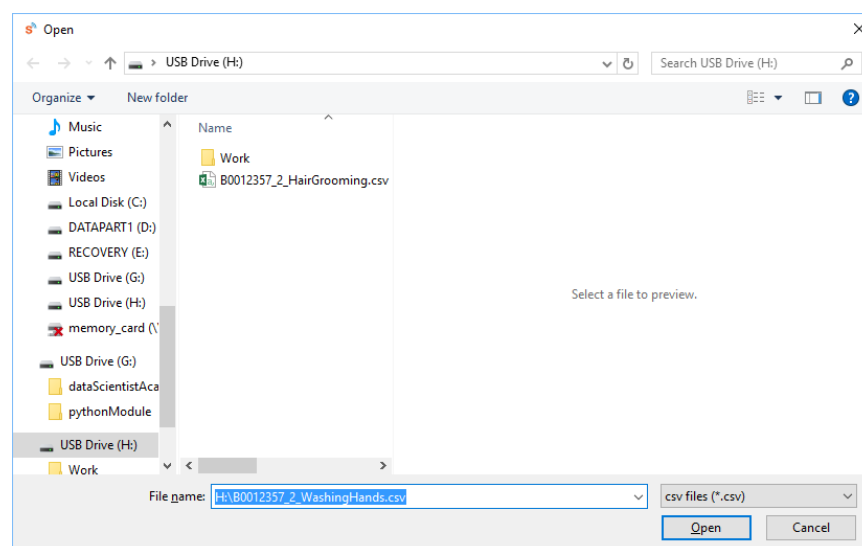


Fig. 7. Uncheck and Recheck the Save to CSV button. Provide a New filename

7. Repeat the data collection steps for the **Second and Third activity**. Making sure you name the files appropriately.
8. **Data Collection is now complete.** You should Check that you have **4 separate files**, 1 for the calibration data and 1 for each of the 3 activities. These files should be appropriately named and structured.

DATA SUBMISSION GUIDANCE

This section describes how to upload your data and submit it through the TurnItIn.

Submission Preparation

A text-based readme files should accompany the submission of your collected data. The readme should briefly describe the data collected, verify the activities collected, the hand / wrist to the shimmer was attached, the duration of the recording as well as to confirm the details regarding the sampling rate and sensitivity set for the Shimmer used.

Uploading Data

The submission system **can only accept a single file upload** therefore, at this point, the readme text file **and** four separate.csv recording should be compressed into a single .zip file, named appropriately as prescribed previously in this guide.

The data should be submitted via the Data Collection Dropbox link (Figure 7), located within the Assignment 2 Folder that can be accessed from the Assignments 2017 Link, present on the module contents page. Within this folder, you should click on the TurnItIn Dropbox Link and follow the onscreen guidance.

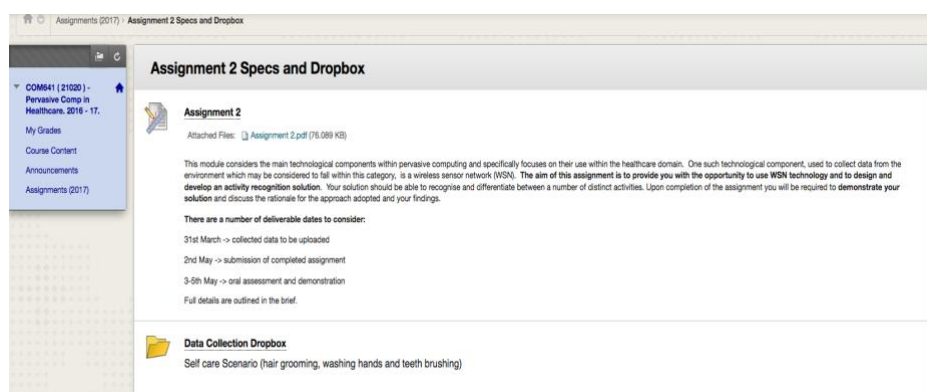
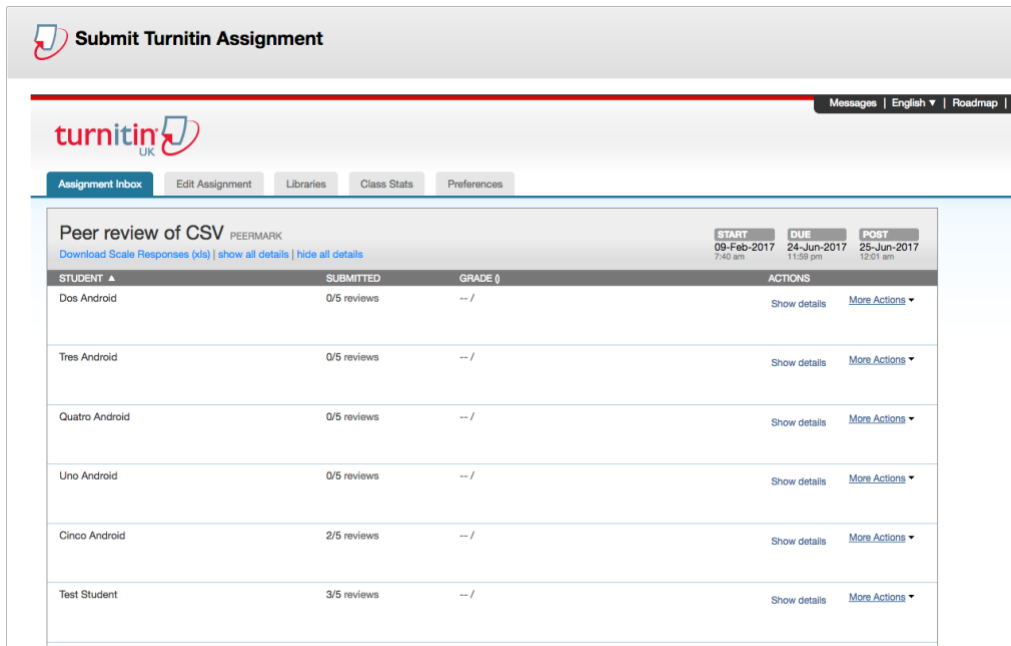


Figure 7. Collected Data should be submitted via the Collection Dropbox.

Once you've uploaded the approach .zip file via the TurnItIn interface, it is essential that you click the Submit button to process the submission.

Post Submission and Accessing Other Datasets

Following the submission of your work, staff will create a **second TurnItIn dropbox** within the Assignment 2 Folder. Here, you will be provided with access to datasets collected by 20 other students who were working within the same data collection scenario. This access is granted via the Peer Review feature within TurnItIn. Upon accessing the Peer Review Dropbox, you will be presented with a list of available submissions (Fig 8). For each submission, you will be able to access and download the respective .zip file (Fig 9).



The screenshot shows the Turnitin interface for a 'Peer review of CSV' assignment. The top navigation bar includes 'Submit Turnitin Assignment', 'Messages', 'English', and 'Roadmap'. Below this, the 'Assignment Inbox' tab is active, showing a table of student submissions. The table has columns for 'STUDENT', 'SUBMITTED', 'GRADE', and 'ACTIONS'. The 'ACTIONS' column contains links for 'Show details' and 'More Actions' for each submission.

STUDENT	SUBMITTED	GRADE	ACTIONS
Dos Android	0/5 reviews	-- /	Show details More Actions
Tres Android	0/5 reviews	-- /	Show details More Actions
Quatro Android	0/5 reviews	-- /	Show details More Actions
Uno Android	0/5 reviews	-- /	Show details More Actions
Cinco Android	2/5 reviews	-- /	Show details More Actions
Test Student	3/5 reviews	-- /	Show details More Actions

Figure 8. The Peer Review System in TurnItIn will list available student dataset submissions.

Appendix 2



Pervasive Computing in Healthcare

Lab Class week 10

Data Cleaning and Wrangling

Data cleaning and wrangling

This week, the lab class will cover concepts surrounding data validation/ cleaning and data aggregation/ wrangling. You will be provided with a number of datasets that will support you to gain experience in data aggregation and data validation.

Sources of Error

During the data collection process, a number of errors may be introduced into the data that either require the data to be cleaned (i.e. some sections of the data must be removed or the signal filtered) or that the data must be discarded. You will more than likely come across many of these errors as you aggregate data from various other student data collection submissions.

Common sources of error within activity recognition experiments include:

- Incorrect activity name/ labelling
- Poor calibration/ not calibrated data
- Incorrect / inconsistent orientation of the sensor on the body
- Incorrect / inconsistent location of the sensor on the body
- Additional sources of noise within the signal i.e. data not reflective of the activity being performed
- Data clipping due to the wrong Sensitivity calibration being used.

Data Collection Scenario

The data files contained within the practical folder on Blackboard contain accelerometer data collected from Shimmer whilst a person completed a number of activities. Specifically, walking, descending stairs, climbing stairs and standing activities were undertaken. The data collection protocol surrounding this experiment specified that data should be collected from a Shimmer using a sampling frequency of 51.2Hz and a Sensitivity of $\pm 4g$. The Shimmer was to be placed on the left wrist in the orientation shown in Figure 1. Prior to data collection, the Shimmer should have been calibrated. Note that the orientation of the device on the body is such that if the participant stands still, with their arms at their side, the gravitational vector will be sensed in -Y direction, meaning the signal will vary around -9.87 (Approx. $10m/s^2$)



Fig 1. The proposed placement and axis orientation of the shimmer.

Data files

The data from each of the activities was saved to separate .csv files. The type of activity is reflected in the file name. Each file contains raw and calibrated accelerometer data, plus the timestamp, milli-seconds and raw units. In total, each file therefore contains 8 columns. The raw and calibrated data can be identified using the header row. Columns 6-8 represents the calibrated accelerometer data for the X, Y and Z axis, respectively.

Practical Lab Tasks

The following sections outline the processing steps for obtaining, cleaning and wrangling data.

Obtaining and cleaning data

- Download and unzip the data from blackboard.
- For each of the files, plot the “**calibrated data**” in Excel or Matlab. Take a screenshot of the data and save this for future reference.
- Using the Excel plot of the calibrated data for each file and your developing knowledge of accelerometry data, determine whether or not the data under investigation is usable or not.
 - In cases where you can clean the data (by removing sections of noise, irrelevant segments to the target activity), proceed to do so and then save the cleaned data to file (i.e. #filename_clean.csv).
 - If data cannot be cleaned (i.e. you suspect bad calibration, wrong orientation etc.) then discard the data and do not include it in any further analysis.
- Complete the table below to provide a data summary of the cleaning process for each file.

File Name	Data Quality (Good/Poor)	Comments on Data Cleaning	Decision (Keep/Discard)
<i>Example 1</i>	<i>Good</i>	<i>Noise at end from sample 2564. Data to be trimmed</i>	<i>Keep</i>
<i>Example 2</i>	<i>Poor</i>	<i>Data not calibrated correctly</i>	<i>Discard</i>
Walking 1			
Walking 2			
Walking 3			
Walking 4			

Upstairs 1			
Upstairs 2			
Upstairs 3			
Upstairs 4			
Standing 1			
standing 2			
standing 3			
standing 4			
Downstairs 1			
Downstairs 2			
Downstairs 3			
Downstairs 4			