Faculty of Engineering

BEng in Engineering Design

YEAR 4 PROJECT

AI-Enhanced Anomaly Detection in Supported Living Environments

Andre du Plessis

Project thesis submitted in support of the degree of
Bachelor of Engineering

Project Advisors:
Professor Theo Tryfonas, Civil Engineering
Dr Paul Harper, Civil Engineering

May 2023

Executive Summary

This report aims to address the growing demand for social care in the UK due to an ageing population and the increasing prevalence of neurodiverse conditions. Traditional care facilities may struggle to accommodate this demand, making alternative solutions like supported living an attractive option. Supported living environments offer a more scalable and person-centred approach to care, allowing individuals to enjoy greater freedom, security, and independence. These environments can help reduce isolation, promote emotional well-being, and provide a more comfortable and dignified living experience compared to institutions or residential care homes.

Artificial Intelligence (AI) has the potential to significantly enhance sensing and monitoring systems in supported living situations. By analyzing sensor data from wearable devices, AI models can detect patterns, identify anomalies, and provide insights into residents' behaviour and well-being. This project explores the use of AI models to improve system performance, with a focus on activity detection, and early identification of potential health issues. The implementation of AI-driven sensing and monitoring systems could lead to better care quality and efficiency, as well as reduce the need for constant supervision, allowing for a more efficient allocation of resources.

A comprehensive review of existing literature highlights several challenges and concerns related to smart home monitoring systems, such as privacy, efficiency, and ease of implementation. The report examines the benefits of wrist or hip-worn devices for collecting triaxial and gyroscopic data and the use of dynamic active learning methods for accurately detecting anomalous events. Traditional machine learning methods are chosen over neural networks for efficient and accurate results with limited data samples. An ensemble model is employed to provide an additional layer of robustness in anomaly detection, leveraging the strengths of multiple classifiers.

The methodology consists of five stages: feature extraction and data pre-processing, novelty detection, classification and uncertainty sampling, initializing the anomaly detection model, and assembling and training the ensemble. Publicly available datasets are used to validate the methodology and evaluate the performance of both the classification model and the anomaly detection model. The evaluation criteria include the accuracy scores obtained for a specific number of annotations in the classification model and the false positive rate for the anomaly detection model.

The results demonstrate promising performance in activity classification and anomaly detection, providing a solid foundation for future research. However, several limitations need to be addressed to further improve the system. These include using a consistent data source throughout the entire process, identifying more complex activities, filtering training data for anomalies, evaluating false negatives, and improving the model's generalizability across diverse user activities, patterns, and demographics.

By refining the approach and addressing these limitations, a more accurate and reliable system for detecting anomalies in supported living situations can be developed. This could ultimately contribute to enhancing residents' quality of life and independence while providing timely information for caregivers to intervene when necessary. In turn, this could promote better health outcomes and more efficient allocation of care resources, ultimately helping to address the growing demand for social care in the UK.

Acknowledgements

Acknowledgements I acknowledge Dr Paul Harper and Professor Theo Tryfonas, both of Bristol University for their continued support, guidance, and help throughout this project. Dr Ryan McConville of Bristol University has been invaluable in his knowledge of machine learning systems and human activity recognition. I also acknowledge Michael McCann of MJMcCann Consulting for the useful discussions we have had regarding modelling techniques and strategies.

DECLARATION

The accompanying research project report entitled: AI-Enhanced Anomaly Detection in Supported Living Environments is submitted in the fourth year of study towards an application for the degree of Bachelor of Engineering in Engineering Design at the University of Bristol. The report is based upon independent work by the candidate. All contributions from others have been acknowledged above. The views expressed within the report are those of the author and not of the University of Bristol.

I hereby declare that the above statements are true.

Signed (author) Andre du Plessis

. . . . . .

Full Name Andre du Plessis

. . . . . .

Date 2023/04/27

. . . . . .

Contents

List of Tables

List of Figures

List of Abbreviations

The next list describes abbreviations that are used within the body of the document

Definitions

AI	Artificial Intelligence
AP	Affinity Propagation
ARC	Activity Recognition Chain
BvSB	Best-vs-Second Best
CECS	Combined External Consensus Score
CECV	Combined External Consensus Vote
CICS	Combined Internal Consensus Score
CNDE	Consensus Novelty Detection Ensemble
CNN	Convolutional Neural Netwok
DAL	Dynamic Active Learning
EP	Entropy
FNSW	Fixed-Size Non-Overlapping Sliding Window
FOSW	Fixed-Size Overlapping Sliding Window
HAR	Human Activity Recognition
ICS	Internal Consesus Score
ICV	Internal Consensus Vote
IF	Isolation Forest
KNN	k-Nearest Neighbours
LOF	Local Outlier Factor
LSTM	Long Short-Term Memory
MAD	Median Absolute Deviation
ML	Machine Learning
NN	Neural Networks
OCLuDAL	OCSVM Clustering Dynamic Active Learning
OCSVM	One-Class SVM
RBF	Radial Basis Function
RCE	Robust Covariance Estimation
SVM	Support Vector Machine

1. Aims and Objectives

1.1 Introduction

In 2019/20, 1.9 million new requests for social care were made, of which only 839,000 received long-term care. Overall, about 43% received support, about 28% received signposting or advice and the remaining 29% of the total who applied received no care at all [1]. In the UK, the ageing population is forecast to rise from 19% to 22% of the population in the next ten years [2]. An estimated 2.16% of adults in the UK are believed to have learning disabilities [3] and a further 1.1% with Autism[4]. Adaptations to current social care are required to handle this increase. One of the tools that can be utilised is that of supported living.

Supported living for the elderly and adults with neurodiverse conditions allows these individuals to enjoy greater freedom, security and independence. This can help ensure that they have access to the care and support they need while still maintaining some level of autonomy in their living space. With the right combination of housing, care, and recreational amenities, a supported living environment can be a more comfortable and dignified option than institutions or residential care homes. Additionally, supported living can help reduce isolation as residents are more likely to benefit from social interaction, promoting emotional well-being.

As traditional care facilities may struggle to accommodate the growing demand, alternative solutions, such as supported living, can offer a more scalable and person-centred approach to address the care needs of these individuals.

1.2 Overall Group Aims

A group of students at the University of Bristol is working on this project in collaboration with Coral Living [5], a company specializing in the design and development of sustainable smart homes with a focus on supported living. As an industry supervisor, Coral Living provides guidance and expertise to help the students develop an innovative and affordable housing and recreational space that improves the quality of life for adults with old age, learning disabilities, and/or autism. The project aims to optimize the internal layouts of accommodations, research and evaluate structural solutions, materials, and building services for 2-3 storey structures, and investigate potential sensing and monitoring systems. Minimizing the embodied carbon emitted from these structures is also an important consideration. Additionally, the project should provide spaces for therapies and other activities while accommodating 6-10 people with an external common area and internal social circulation.

While group members focus on various aspects of supported living, such as construction, space optimization, and building services, this project specifically addresses sensing and monitoring systems for residents' safety and well-being. By combining the diverse expertise, a joint report will be produced next year, presenting a holistic approach to designing and implementing an innovative and sustainable supported living environment.

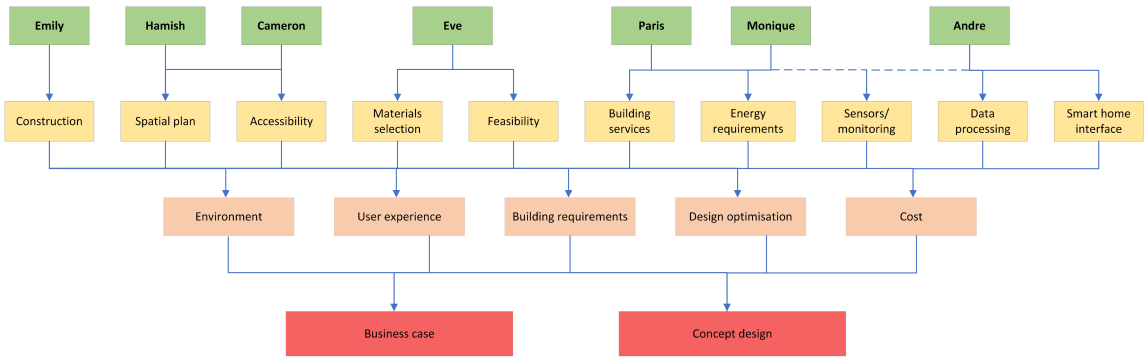

FIGURE 1.1: Group System Diagram
Illustration detailing how the individual projects relate to the group aim

1.3 Individual Aims

Implementing sensing and monitoring systems in supported living situations can increase safety and security for residents, improve the quality of life and independence, and enhance communication and support from caregivers and family members. These systems can provide real-time information about the well-being and activities of residents, allowing for timely intervention in case of emergency or health concerns.

Artificial Intelligence (AI) can enhance sensing and monitoring systems in supported living situations by analyzing sensor data to detect patterns, identify anomalies, and provide insights into residents' behaviour and well-being. In this project, AI models will be explored to improve system performance, focusing on activity and fall detection, and early identification of potential health issues, thereby improving care quality and efficiency.

There is expected to be a shortfall of 160,000 caregivers by 2032 in England [5]. The implementation of monitoring systems can help reduce the need for constant supervision and in-person care, allowing for a more efficient allocation of resources. They can also help identify potential health issues early on, potentially preventing the need for costly medical interventions. Additionally, the use of these systems can improve the overall quality of care, leading to better health outcomes and potentially reducing long-term healthcare costs.

However, implementing AI-driven sensing and monitoring systems in supported living environments comes with challenges such as ensuring data privacy, addressing potential ethical concerns, and developing user-friendly systems that can be easily adopted by residents and caregivers.

1.4 Individual Project Objectives

1. Conduct a comprehensive review of the existing literature on sensing and monitoring technologies used in ambient assisted living environments, focusing on their acceptance by residents, potential challenges in implementation, and the impact of these technologies on the quality of care.

2. Investigate and compare the performance of various AI models for detecting and classifying activities and potential health issues in home environments, using available data sources to determine the most promising techniques for this application. Explore the limitations and biases of the data used for model training and evaluation.

3. Develop a data-driven model that can learn from sensor data to create a baseline profile of the occupant's normal behaviour, including classifying their activities, ensuring that the data sources, preprocessing techniques, and feature extraction methods are appropriate for the project's goals.

4. Apply the model to a set of case study data to evaluate its effectiveness in detecting abnormal behaviour/events.

2. Literature Review

In this literature review, the current state of smart home technologies is discussed, focusing on technology readiness and acceptance, system feasibility, human activity recognition (HAR), and anomaly detection. User concerns and preferences are explored, as well as the limitations and challenges in existing HAR and anomaly detection methods. This review will serve as a foundation for the methodology proposed, in which the aim is to address these challenges and develop an effective and accurate process for detecting concerning behaviour (such as falls, erratic sleep behaviour, excessive toilet trips, or a general lack of activity) from raw data sources.

2.1 Existing Smart Home Technologies

Smart home technologies have been a rapidly growing field of interest, with exploratory surveys for technology acceptance rates and prototype models beginning in the early 2000s [6] and growing increasingly quicker since then. Already, systems such as the Amazon Alexa [7] and Google Home [8] are in full deployment and are capable of many tasks, from controlling the lights and other electronic devices to monitoring the thermostat and automating everything in between.

2.1.1 Technology Readiness and Acceptance

Offermann-van Heek et al.[9] raises the question of what influence users' decisions and opinions hold on the technologies used for assisted living in people's homes. The study uses surveys and models two scenarios of social care requirements; low level needs and moderate needs. It was decided that only the low level care needs were relevant to this paper, described as aged under 71, living alone, having small health problems and being somewhat overtaxed with daily chores. The study proceeded to evaluate the different technologies in emergency detection and medical remainders.

The main concerns of were linked to the invasion of privacy, surveillance, data security and not receiving enough human contact. The overall evaluation of acceptance to new technologies was positive. There was a strong rejection of the usage of camera systems. However, privacy concerns can be somewhat alleviated by anonymising the visual data and giving the user control over turning the system on or off [10]. The most positive feedback and acceptance came from smart watches, followed by emergency buttons and motion detectors.

Bergmann et al. [11] reviewed in greater detail the preferences of patients and clinicians in the design of body-worn sensors. They concluded that the systems should be compact and simple to operate. Their purpose is not to replace care workers but to work alongside them and reduce the number of journeys to hospitals.

2.1.2 System Feasibility

Prototype design for monitoring systems and data flows [12], such as the GiraffPlus project [13], has demonstrated the feasibility of system architectures and middleware components for data gathering. However, the project did not develop methods for processing and analyzing the data in a meaningful way. This limitation is a common issue in the field, as many systems are able to collect data, but struggle to make sense of the results. In order to address this challenge, researchers and developers need to focus on developing advanced algorithms and machine learning models that can effectively classify and analyze complex and dynamic data sets. [14] proposed a mobile and home based monitoring system for fall and activity detection. Although successfully deployed with a working communication and messaging system, they did not provide many meaningful results in regard to activity and fall detection elements.

2.2 Activity Detection

HAR is the process of automatically identifying and classifying human activities, such as walking, running, or cooking, based on data collected from sensors and other monitoring devices [15]. The Activity Recognition Chain (ARC) is the system of actions typically required to perform activity recognition [16]. It involves several steps, including:

- Data acquisition: This involves collecting data from sensors or other sources that provide information about the person's activity.

- Data preprocessing: Involves operations such as unit conversions, sensor calibrations, synchronisation between devices (if more than one is used) and cleaning of corrupted or noisy data. The data can then be segmented and windows formed to be classified.
- Feature extraction: In this step, the relevant features are extracted from the preprocessed data. These features can be crafted.
- Classification: In this step, the extracted features are used to classify the person's activity. This may be done using machine learning algorithms, such as decision trees, support vector machines, or neural networks.

2.2.1 Data Acquisition

There are two main types of data acquisition methods for HAR: visual and sensor-based [17]. Visual methods, which use cameras and other imaging devices, are not commonly used due to privacy concerns [9]. A way around the privacy concerns of visual data is the use of social media-based sensor data, as explored in [18]. A difficulty in this is a lack of unambiguous labels and difficulty for even a human to identify the activities at hand. Sensor-based methods, on the other hand, are more widely used and can be further divided into two categories: wearable devices and dense sensing networks [19].

Wearable devices, such as smartwatches, fitness trackers, and accelerometers, are small, portable devices that can be worn by individuals to collect data on their movements and activities. These devices typically use a combination of sensors, such as accelerometers, gyroscopes, and heart rate monitors, to track and record data. Dense sensing networks, on the other hand, are more complex systems that use multiple sensors, such as RFID tags and motion sensors, to monitor and classify activities.

After data has been acquired from sensors and monitoring devices, the next step in the HAR process is classification. This involves using algorithms and machine learning techniques to analyze the data and identify specific activities. There are two main approaches to classification: traditional machine learning (ML) algorithms and neural network (NN) models [15].

2.2.2 Data Preprocessing

Data segmentation is the process of dividing a large dataset into smaller, more manageable segments. This is often done to make analysis and interpretation of the data easier and more efficient. A method of this is the sliding window that divides the dataset into a series of fixed-length "windows". These windows can be overlapping or not [19]. They continue to illustrate another consideration in the fusion of multimodal sensor data. Data can be fused at different levels; data level fusion, feature level fusion or decision level fusion. [20] investigates a number of fusion techniques when used in convolutional neural networks, concluding that late or hybrid fusion techniques outperform earlier fusions.

2.2.3 Feature Extraction and Classification

Features can be engineered by hand, comprising various statistical metrics in both the time domain, such as the mean and standard deviation [21] or the frequency domain, properties such as coherence and entropy. The alternative is through automatic learning methods found in artificial neural networks.

Traditional Machine Learning

Traditional machine learning algorithms, such as decision trees [22], support vector machines (SVM) [23], hidden markov models [24] and k-nearest neighbours (KNN) [25], use a set of pre-defined rules and patterns to classify data and often require feature engineering to get accurate results. While these algorithms have been successful in strictly controlled environments and with limited input data, they have several limitations that make them less effective in real-world scenarios. For example, these algorithms often require multiple pre-processing steps and hand-crafted features, which can be time-consuming and inefficient. Additionally, the use of shallow features often leads to poor performance on incremental learning or unsupervised learning tasks. These limitations highlight the need for more advanced algorithms and machine learning models that can handle complex and dynamic data sets without the need for extensive pre-processing and feature engineering [26], [27].

In a study published in 2021 [28], researchers compared the performance of random forest [29] and convolutional neural network (CNN) algorithms [30] when running on low energy microcontroller devices. They found that the machine learning methods were more energy efficient [31] and suitable for wearable devices. However, these

methods may not be suitable for long-term analysis, such as anomaly detection, where the advantages of energy efficiency are lost.

Neural Networks

Neural networks are a type of machine learning algorithm that are modelled after the structure and function of the brain. They are composed of a large number of interconnected nodes, called neurons, which are arranged in layers. These layers of neurons are connected by weights, which are learned by the model during training [32].

One of the key advantages of neural networks is that they are able to automatically learn features from the data, rather than requiring the data to be manually transformed into a set of features. This makes them useful for tasks where the underlying patterns in the data may be difficult to identify or describe. Common architectures used in activity detection are CNNs [33], [34], and long short-term memory networks (LSTMs) [35], [36]. LSTMs are able to capture long-term dependencies in data by using gating mechanisms to control the flow of information through the network. These gating mechanisms allow the LSTM to retain information for longer periods of time, and to selectively forget irrelevant information. However, they require significant amounts of labelled data to draw meaningful conclusions [37], further increasing the costs of human labour.

Learning Methods

An important consideration in making the classification decision is in the type of data available: labelled on unlabelled.

In supervised learning, the system is trained on labeled data, where the correct activity for each data point is known. In this case, the classification technique should be able to learn the mapping between the input data and the corresponding labels, so that it can accurately predict the activity for new data. Most methods of activity detection utilise supervised techniques [38], [39]

In unsupervised learning, the system is not provided with labelled data and must learn to recognize activities by discovering patterns in the data. In this case, the classification technique should be able to identify clusters or groups of data points that share similar characteristics and use these clusters to classify new data. [40] proposed a method of applying unsupervised learning to activity recognition.

In semi-supervised learning, the system is provided with a small amount of labelled data and must use this to learn to recognize activities in the remaining unlabeled data. In this case, the classification technique should be able to learn from the labelled data, and then use this knowledge to classify the unlabeled data.

Dynamic active learning (DAL) is a machine learning technique that focuses on continually selecting and labelling the most informative and valuable data points during the training process. This can help improve the efficiency and effectiveness of the training process by allowing the model to focus its learning on the most useful data. Novelty detection algorithms, such as One-Class SVMs (OCSVM) [41] can be used to quickly identify a range of activities for minimal annotations. In contrast to traditional active learning methods, dynamic active learning also allows for the discovery of unseen patterns and activities in the data. Additionally, a novel sample selection policy that considers the uncertainty, diversity, and representativeness of samples can be used to identify the most valuable data points. The methodology proposed in [42] proves to be equally successful with other approaches but with a fraction of the available annotations. This method has shown promise in handling unlabeled data and incremental learning tasks, making it well-suited for real-world applications. As such, this approach to learning will be used for the activity classification in the methodology. However, it does not carry the model forward into anomaly detection.

2.3 Anomaly Detection

Anomaly detection is built as a final step on the ARC model as performing additional analysis on the behavioural model that has been derived. It refers to the detection of 'abnormal data' that can indicate a change in user behaviour and therefore, potentially alert carers of the need for some form of intervention (e.g. a call to check that the resident is okay). Some contextual examples could be not getting out of bed in the morning or forgetting to cook dinner; both indicators of a change of mood or behaviour.

It's typically done through profiling and/or discriminating [19]. Profiling involves building a model based on historical data and comparing incoming data to this model. Any data points that fall outside the acceptable margins are flagged as anomalous behaviour. This method is versatile as it can handle unseen anomalies. Discriminating,

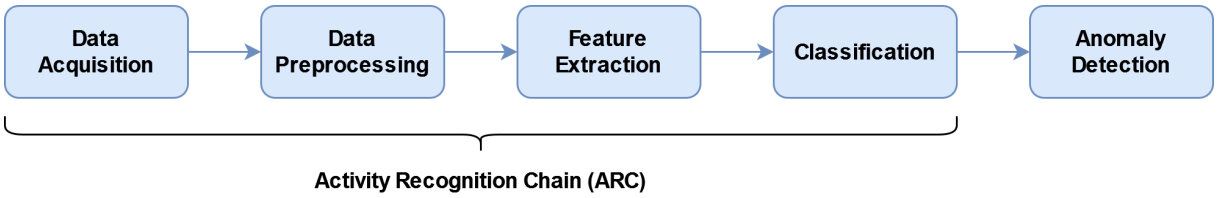

FIGURE 2.1: Activity Recognition Chain and Anomaly detection visualisation

on the other hand, involves matching incoming data against historical known anomalies to flag behaviour. While this method is not as robust, it can be useful in identifying common causes of emergencies such as falls or fires.

Anomalies can be classified into three main categories: point, contextual, and collective anomaly [43], [44]. Point anomalies are individual data points that deviate significantly from the expected norm. For example, a single instance of a high heart rate or a sudden change in room temperature would be considered a point anomaly. Contextual anomalies are anomalies that are only apparent when considered in the context of other data points. For example, lying down for the duration of the night is normal, however if they are in the kitchen or bathroom then it would be considered a contextual anomaly. Collective anomalies are collections of similar data points that can be considered abnormal together when compared to the rest of the data. For example, a consecutive 5-day spell of altered sleep patterns would be considered a collective anomaly.

Another method of identifying anomalous behaviour is explored in [45], where rather than identifying singular anomalous activities, entire days are classified as anomalous if the number of activities performed deviates from a normal distribution. However, this does not provide much insight as to the cause of the anomalies.

An issue in HAR and so inherently also in anomaly detection is the variation in how people behave. [42] introduces the term patterns to describe the three dimensions of variation; diverse daily activities, different people performing the same activities differently, and the same person performing the same activity differently when in a different mental state. An adaptive anomaly detection system is proposed in [46] that allows the human user to provide their own feedback as to whether the system is providing the correct predictions. The model also incorporates data ageing and forgetting factors to increase the adaptiveness of the model. This system of detection is built on an ensemble approach proposed in [47], where a network of different outlier detection models are used to flag anomalous sleep behaviour. A high level of accuracy and transparency in its results can be seen. However, it relies on pre-existing labelled data to work and suggests no method on reaching this point.

2.4 Conclusion of Literature Review

In conclusion, the literature review addresses several challenges and concerns related to smart home monitoring systems, with a particular focus on privacy, efficiency, and ease of implementation. The benefits of a wrist or hip-worn devices have been identified, which offer advantages in terms of privacy and public dataset availability, as well as simple installation. These devices collect triaxial and gyroscopic data, which serve as the raw input for the system.

The use of a dynamic active learning method is a key component of accurately detecting anomalous events, achieving high accuracy with a minimal number of data samples. This decision was driven by the high cost of annotation, which makes it essential to prioritize informative sampling. The active learning method involves pre-processing the data, segmenting it, and extracting relevant features. By adopting this approach, not only are the limitations of traditional machine learning algorithms addressed but also ensures that the methodology is resource-efficient and effective.

The choice to use traditional ML methods over neural network models was informed by the need for efficient and accurate results with limited data samples. While neural networks have shown great promise in various applications, they often require large amounts of training data and may not be suitable for the active learning approach. By focusing on feature extraction and engineering, a balance can be achieved between computational efficiency and classification accuracy, making the system more suitable for real-world smart home monitoring applications.

Furthermore, the ensemble model employed for the anomaly detection provides an additional layer of robustness by leveraging the strengths of multiple classifiers. This helps to improve the overall accuracy of the system and ensures that anomalies are effectively detected and monitored.

3. Research Methodology

In the preceding literature review, several key objectives were considered relevant to the study. The review provided a comprehensive analysis of sensing and monitoring technologies used in ambient assisted living environments, focusing on their acceptance by residents, potential challenges in implementation, and their impact on the quality of care. In addition, there was a discussion of the importance of data sources, preprocessing techniques, and feature extraction methods for developing a data-driven model that learns from sensor data to create a baseline profile of an occupant's normal behaviour.

3.1 Methodology Overview

This section introduces the process developed to identify anomalous behaviour using accelerometer data and machine learning techniques. This section aims to illustrate the underlying processes and explain conceptually the transformations in data. Validation of these processes is included throughout Chapter 4.

A systematic diagram of this process can be found in Figure 3.1 The first half of the process focuses on activity classification, utilizing key components of the OCSVM Clustering Dynamic Active Learning (OCLuDAL) method proposed in [42]. This method was chosen because it achieved high performance with a limited number of annotations and was capable of discovering a greater number of classes or activities given a fixed amount of annotated data.

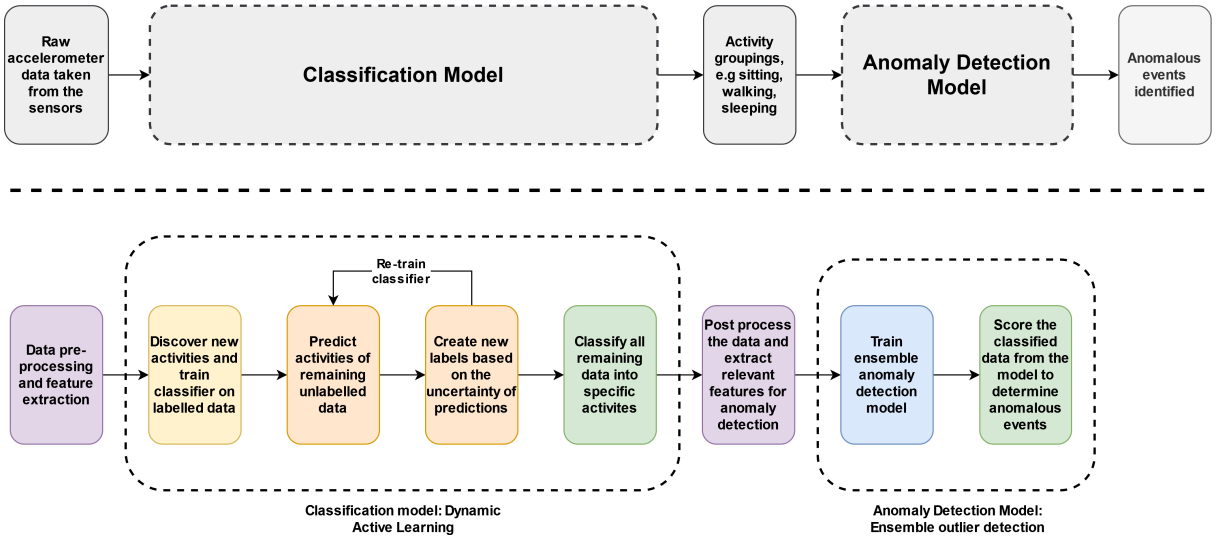

FIGURE 3.1: Overall system architecture

The process can be split into two main models: A classification model is responsible for converting raw numeric data into a set of distinct activities such as walking, standing, and watching TV. The other is an anomaly detection model that uses the activity groupings as input and scores them to identify potential anomalies in behaviour. The two flows on either side of the dashed line describe the same process but in different levels of detail.

The OCLuDAL method employs a dynamic active learning approach, which addresses the costly nature of annotating large datasets. This is achieved by conducting novelty detection and uncertainty sampling iteratively to select the data points that provide the most valuable information when annotated. The second half of the process is dedicated to anomaly detection, utilizing the Consensus Novelty Detection Ensemble (CNDE) method proposed in [47]. This method was selected due to its focus on identifying abnormalities in sleep patterns and its superior performance compared to other anomaly detection approaches. The CNDE method employs an ensemble approach to construct a normality model of an occupant, which is then used to predict outlying new observations. A brief overview of each main process stage is provided, followed by more detailed explanations of the model's development and validation in the subsequent sub-sections.

Stage 1 - Feature Extraction and Data pre-processing

In the initial stage of the process, continuous time series accelerometer data, gathered from a wearable device, is segmented into finite time windows. Feature extraction is the process of calculating the properties of a given set of data, such as mean, max and min. These features are extracted from each dimension of the data and resulting values are standardised. A random subset of data is selected for annotation.

Stage 2 - Novelty Detection

In the next stage of the process, numerous outlier detection models are applied to each unique activity found in the initial annotation. The data points that exist as outliers to all of the models are identified as novel points. An unsupervised clustering algorithm is used to group these novel points and identify characteristic points for annotation. This process is repeated iteratively.

Stage 3 - Classification and Uncertainty Sampling

The labelled data is used to train a classifier, which is then used on the unlabelled data to predict their activities. An uncertainty-based sampling method is used to select the most uncertain samples, which are chosen for further annotation. This method iteratively selects the most informative samples to maximise the increase in classification accuracy for as few annotations as possible. When a maximum amount of annotations has been made, the final classifier can be used to predict the activities of the remainder of the unlabelled data.

Stage 4 - Initialise Anomaly Detection Model

Once the activities have been classified, they are grouped together as observations and metadata are extracted. This metadata can be used to identify patterns and trends within the activities such as durations and days of occurrence etc. The data is then split into training and testing data sets to prepare for the next step of the process. The anomaly detection model is an ensemble composed of subsidiary, independent outlier detection models, each working off of a different algorithm for scoring the observation. Each of these smaller models is trained from the training data.

Stage 5 - Assembling and Training Ensemble

The model is built by connecting different subsidiary models together to form a combined network as described in [47]. The network makes decisions about outliers and anomalies based on an internal (within each subsidiary) and external (between subsidiaries) voting and scoring scheme. To improve the performance of the model, disparities between internal and external votes are used to adjust the score weightings of the subsidiary models. The scores are then recalculated based on the updated weightings. Finally, a normality score and threshold are calculated from the mean and standard deviation to identify any anomalies.

3.2 Stage 1: Feature Extraction and Data Pre-processing

The data is segmented into finite chunks using the sliding window technique. Two types of sliding windows can be used, fixed-size non-overlapping sliding windows (FNSW) and fixed-size overlapping sliding windows (FOSW). FNSW ensures independence in training and prevents overfitting of data. Optimal time windows, T_w, between 4 and 10 seconds are shown in [48].
Consider the raw temporal dataset:

$$\tilde{Y} \equiv \{\tilde{y} \in \mathbb{R}^{T_n \times D_r}\} \tag{3.1}$$

where T_n is the number of individual time entry stamps and D_r is the dimension of the data, e.g acceleration or gyroscope in x, y, z. The number of window samples, W_s can be calculated using the frequency, f in 3.2 below.

$$W_s = T_w * f \tag{3.2}$$

The segmented dataset can now be represented as:

$$\tilde{\mathbb{Y}} \equiv \{\tilde{y} \in \mathbb{R}^{N \times D_r \times W_s}\} \tag{3.3}$$

where N is the total number of segments. It is worth noting that each data point is three-dimensional at this stage.
Feature extraction is then applied to create a set of data points, with each data point consisting of feature calculations for each dimension of data.
The extracted data, $\tilde{\mathbb{X}}$ set can now be represented as:

$$\tilde{\mathbb{X}} \equiv \{\tilde{x} \in \mathbb{R}^{N \times F_n}\} \tag{3.4}$$

where F_n is the number of features multiplied by the number of dimensions, D_r. This transformation of data can be visualised in figure 3.2.

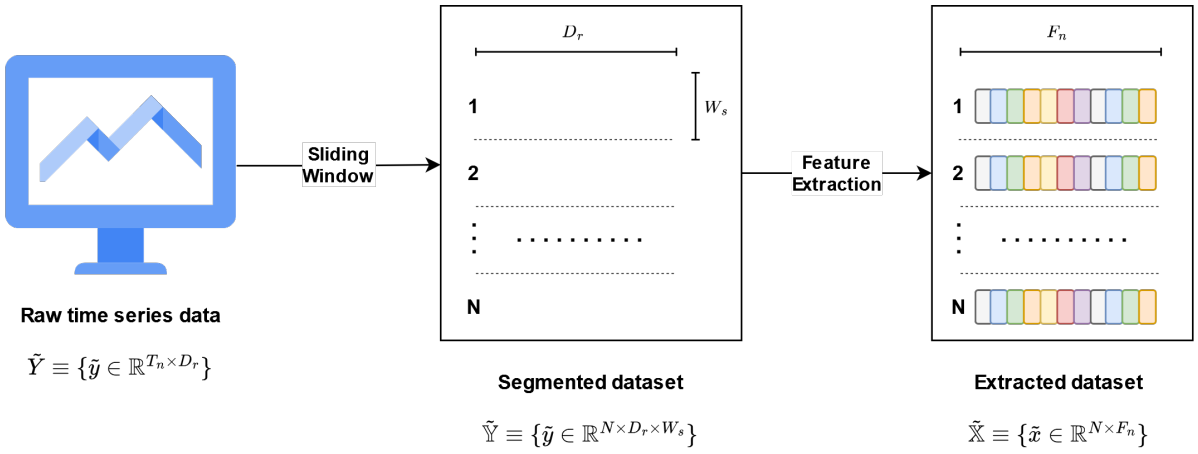

FIGURE 3.2: Segmentation and Feature Extraction
A visualisation of the steps taken to pre-process the raw dataset before classification can take place.

The features extracted, as suggested in [42], are shown in the table 3.1 below.

TABLE 3.1: Activity Classification Feature Extraction

Feature	Description	Formula
F_{mean}	Mean of the data points	$\sum_{i=1}^{n} x_i/n$
F_{min}	Minimum of the data points	$min(x_i)$
F_{max}	Maximum of the data points	$max(x_i)$
F_{sum}	Sum of the data points	$\sum_{i=1}^{n} x_i$
F_{prod}	Product of the data points	$\prod_{i=1}^{n} x_i$
F_{std}	Standard deviation of the data points	$\sqrt{\sum_{i=1}^{n}(x_i-\mu)^2/n}$
F_{mc}	Mean crossing of the data points	$\sum_{i=1}^{n-1} \frac{1}{2}\lvert x_i - x_{i+1}\rvert$
F_{zc}	Zero crossing of the data points	$\sum_{i=1}^{n-1} \lvert x_i - x_{i+1}\rvert$
F_{iqr}	Interquartile range of the data points	$\frac{1}{2}(x_{75} - x_{25})$
F_{skew}	Skewness of the data points	$\frac{\sum_{i=1}^{n}(x_i-\mu)^3/n}{\sigma^3}$
F_{25}	25th percentile of the data points	x_{25}
F_{75}	75th percentile of the data points	x_{75}
F_{kr}	Kurtosis of the data points	$\frac{\sum_{i=1}^{n}(x_i-\mu)^4/n}{\sigma^4}$
F_{seg}	Segmentation of the data points	$\sum_{i=1}^{n} x_i^2$
F_{sep}	Separation of the data points	$-\sum_{i=1}^{n} x_i^2 \log(x_i^2)$

It is important to note that SVM kernel results are negatively affected by differences in magnitude and it is recommended that values are scaled to roughly the normal distribution [49]. The data is transformed by subtracting the mean from every value and scaling by dividing by the standard deviation.

A random subset of data points is chosen for annotation, splitting the $\tilde{\mathbb{X}}$ into labelled dataset $\mathbb{L}$ and unlabelled dataset $\mathbb{U}$. As this selection is random, the more samples that are chosen, the less efficient the final selection of annotated samples will be. However, fewer samples selected at this stage will result in a greater number of novel data points to be used in affinity propagation. This could lead to affinity propagation failing to identify the most representative samples.

For the purposes of demonstrating the classification tasks of Stages 1, 2 and 3; a dummy dataset has been produced to graphically illustrate the development over each stage. The simulated dataset is a greatly simplified version with only two features (x and y) to visualise on a 2D plot. An example of the initial unlabelled and labelled datasets can be seen in Figure 3.3

This process can be summarised in algorithm 1 below:

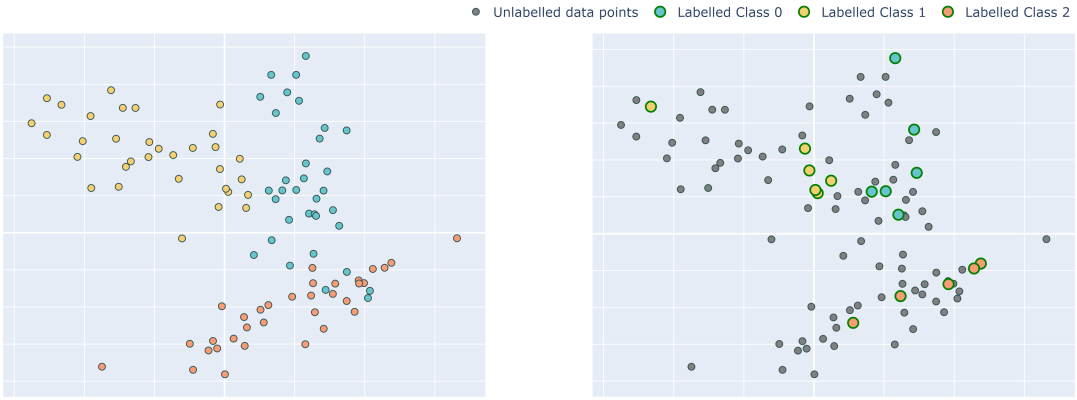

FIGURE 3.3: Dummy Dataset: Initial Annotations
A simulated dataset has been created to demonstrate the activity discovery and sampling selection stages. The dummy set comprises of only 2 dimensions and 3 activities, in contrast to the 90 dimensions and 12 activities of the real dataset. On the left is the entire dummy set with all activities colour coded. On the right is the result following the initial random annotations; grey represents the unlabelled dataset and coloured samples represent the labelled dataset.

Algorithm 1 Feature extraction

Input: $\tilde{Y}$ ▷ Raw temporal dataset
Output: $\mathbb{U}, \mathbb{L}$ ▷ Unlabelled, labelled datasets
$\tilde{\mathbb{Y}} \leftarrow \text{SlidingWindow}(\tilde{Y})$ ▷ Segmented dataset
$\tilde{\mathbb{X}} \leftarrow \text{FeatureExtraction}\ (\tilde{\mathbb{Y}})$ ▷ Transformed dataset
$\mathbb{U}, \mathbb{L} \leftarrow \text{Annotation}(\tilde{\mathbb{X}})$

3.3 Stage 2: Novelty Detection

The main aim of the novelty detection stage is to discover as many of the different activities as possible to increase the classifier's performance in the early stages of the algorithm. The validation for the neccesity of this step and the fine-tuning of the algorithm can be found in section 4.2.3

The first step is to iterate through the unique activity labels identified in the initial annotation and train a novelty detection model to identify novel points. For this, a One-Class Support Vector Machine (OC-SVM) is used, as described in [50]. This unsupervised learning technique aims to minimize the hypersphere of the single class of data points being used to train it. The resulting model creates a boundary; all points outside that barrier are considered outliers or novel points, and all those within are inliers. By training an OC-SVM on each known class from the labelled dataset, it is possible to detect data points that exist as outliers to all the classes. This set is called the conjoint novel dataset, $\mathbb{C}$, and is a subset of the total unlabelled dataset $\mathbb{U}$. A demonstration of the different OC-SVMs applied to the dummy dataset can be seen in figure 3.6.

Affinity propagation (AP) is then used as an unsupervised clustering algorithm to cluster these data points. This algorithm measures the similarities between exemplars, and resulting clusters by sending messages between pairs of samples until a solution has converged.

Unlike other clustering methods, the number of classes does not need to be specified beforehand. Additionally, the exemplars are real data points instead of hypothetical centres, identifying important points for labelling. However, a disadvantage of this technique is the time complexity, which scales on the order of $O(N^2T)$, where N is the number of samples and T is the number of iterations until convergence. This complexity can become problematic for large datasets and is why the trade-off in stage 1 about the number of random points chosen for selection exists.

Two important hyperparameters in affinity propagation are the preference and damping factor. Preference controls the number of exemplars, while damping affects the messaging between data points, creating a trade-off between responsiveness and unwanted numerical oscillations [41]. To evaluate the quality of clustering for a given set of hyperparameters, the silhouette score is recommended in [42]. This metric was introduced in [51] and measures how similar a data point is to its own cluster compared to other clusters and varies from -1 to 1. A score of 1

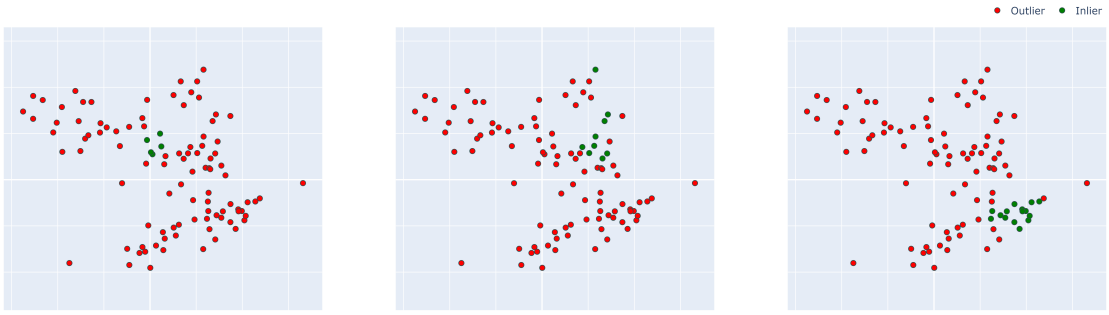

FIGURE 3.4: Novelty Detection with One-Class Support Vector Machines
A demonstration of the three OC-SVMs applied to the dummy dataset. Each is trained on a different activity from the labelled data with the aim of identifying outliers to all of the currently labelled datasets. This identifies all of the datapoints that cannot be explained from the current set of labelled data.

indicates high-quality clustering and -1 indicates low quality. 0 indicates overlapping clusters. The equation is shown in equation 3.5 below.

$$s_i = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{3.5}$$

where b is the similarity score compared to the nearest cluster.

The centers of the resulting clusters, the exemplars, are then chosen for the representative sample set, $\mathbb{S}$, that will be the most useful to annotate. This process of selection is demonstrated in the plot below:

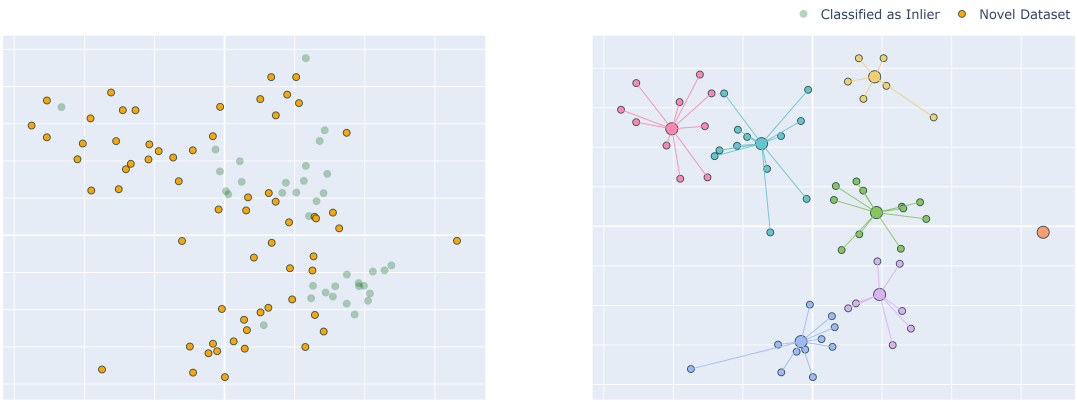

FIGURE 3.5: Novelty Detection with Affinity Propagation
A demonstration of the unsupervised clustering algorithm applied to the conjoint novel dataset, $\mathbb{C}$ using the dummy dataset. On the left is a plot of all the data points identified as outliers by all three OC-SVMs. On the right is the result of the clustering algorithm, wherein each cluster is colour-coordinated and the centres are the larger points. The cluster centres are chosen as the sample novel set, $\mathbb{S}$, to be annotated.

These exemplars are then annotated and the labelled dataset is updated. The resulting labelled and unlabelled dataset from the dummy dataset is illustrated below.

This whole process is repeated several times until certain annotations have been made or other termination criteria have been met. The algorithm for stage 2 can be seen below:

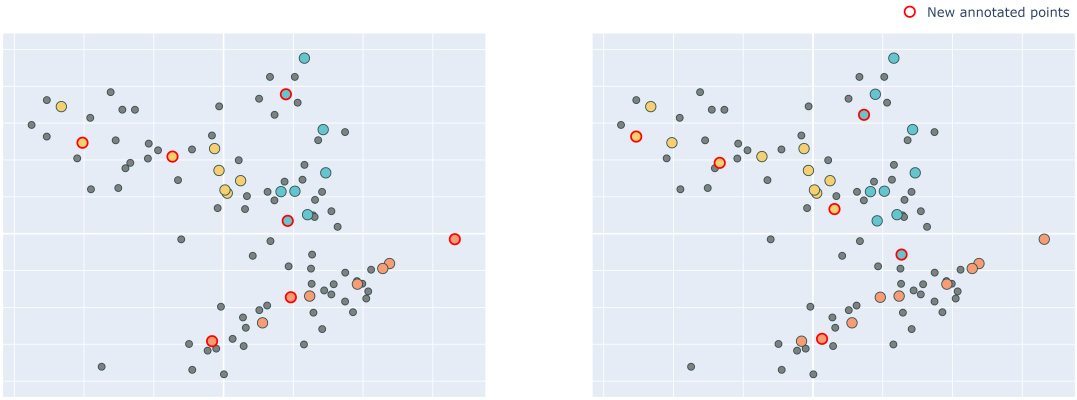

FIGURE 3.6: Novelty Annotation
Following identification by the AP clustering, the exemplars are annotated and the labelled and unlabelled datasets are updated. On the left is a visualisation of the dataset after one full iteration. On the right is after another whole iteration. The new labelled points are indicated with a red border.

Algorithm 2 OCluDAL Step 2: Novelty detection

Input: $\mathbb{U}, \mathbb{L}$ ▷ Unlabelled, labelled datasets
Output: $\mathbb{U}, \mathbb{L}$
  $i = 1$ ▷ Iteration
  while $i < I$ or stopping criterion not satisfied **do**
    $\mathbb{C}_n \leftarrow \text{OneClassSVM}(\mathbb{L}, \tilde{\mathbb{X}})$ ▷ Conjoint novel dataset
    if $\mathbb{C}_n =$ nothing **then**
      Break
    end if
    $\mathbb{S}_n \leftarrow \text{AffinityPropagation}(\mathbb{C}_n)$ ▷ Sample novel dataset
    $\mathbb{L}_n \leftarrow \text{Annotation}(\mathbb{S}_n)$ ▷ Labelled novel dataset
    $\mathbb{L} \leftarrow \mathbb{L} \cup \mathbb{L}_n$ ▷ Updated labelled dataset
    $\mathbb{U} \leftarrow \mathbb{U} \backslash \mathbb{L}_n$ ▷ Updated unlabelled dataset
    $t \leftarrow t + 1$
  end while

3.4 Stage 3: Classification and Uncertainty Sample Selection

In this stage, a classifier is trained on the labelled dataset obtained from the novelty detection step. The main classifier used for high-dimensional data is a Support Vector Machine (SVM). SVM is a supervised learning algorithm introduced in [52] and [53]. It aims to find the optimal hyperplane that separates two classes in a high-dimensional feature space. The training data is mapped to a higher dimensional space using a kernel function, which allows for non-linear separation between classes.

There are several kernel functions available for SVM, including linear, polynomial, radial basis function (RBF), and sigmoid. Among these kernels, the RBF kernel is recommended in [49] as it can handle non-linear relationships between class labels and attributes. It also has fewer hyperparameters than the polynomial kernel, making it easier to tune.

After obtaining the SVM model from the labelled data, the classifier can predict the labels of the unlabelled data. However, instead of focusing on the resulting classification, the interest is in the probabilities associated with each prediction. The most informative samples for annotation are the ones that the classifier is most uncertain about. This process is known as uncertainty sampling [54].

An alternative option is the k-Nearest Neighbours classifier as described in [25]. It is a simple non-parametric supervised learning technique that determines the classification of a datapoint based on the average values of its k-nearest neighbours. The value of k is highly data-dependent and a specific value may not perform as well on different sets of data. In general, a high k-value suppresses noise but blurs the boundaries between classes.

A validation of the final classifier used and its hyperparameters can be found in section 4.2.4.

Uncertainty sample selection is a process of selecting samples that the classifier is most uncertain about annotating. By selecting the most uncertain samples, the borders between neighbouring similar activities can be better defined for the classifier to make predictions. In [55] for image classification and [42] for activity recognition, two methods for measuring the information gain from a single annotation are explored, namely entropy measure (EP) and best vs second best (BvSB). Consider a sample x_i, whose classification is Y based on its class probabilities, described as a distribution $P \in \{p_i, p_1...p_k\}$ where k is the number of classes.

EP is based on the concept of entropy, which measures the overall uncertainty across all classes. The discrete entropy of a single classification, $H(Y)$ can be calculated by:

$$H(Y) = -\sum_{i=1}^{k} p_i log(p_i) \tag{3.6}$$

A classifier will be uncertain if a data point has a distribution of high entropy. However, the entropy distribution and score can be skewed by probability values of lower probability, unimportant classes, as shown in figure 3.7. Since the interest is in the misclassification and uncertainty between activities, the greedy approach, BvSB, is proposed. This method only considers the first and second most probable classifications for each data point, and scores it based on the probability distance between these two classifications. The S_{BvSB} is therefore calculated by:

$$S_{BvSB} = p_{best} - p_{secondbest} \tag{3.7}$$

All data points are ranked by this score, and the lowest-scoring, most uncertain points are selected for further annotation.

The labelled dataset is updated with the newly annotated data, and the data points that were selected for annotation are removed from the unlabelled set. The entire process of active learning is repeated until a termination criterion is met, such as a maximum number of iterations or a certain amount of time. The number of samples selected for annotation in each iteration plays a crucial role in the effectiveness of the process. If too many samples are selected at once, each sample may become less informative, and the effectiveness of subsequent selections may be reduced. On the other hand, selecting too few samples may require more iterations, which may increase the computational cost of training the SVM. Thus, there is a trade-off between the cost of annotation and the computational cost of training, and an optimal number of samples should be selected in each iteration to balance these factors.

Once the finalised $\mathbb{L}$ has been established, it is used to train a final SVM model. It is with this model, that the remainder of $\mathbb{U}$ is classified to create of classified accelerometer data, $\tilde{\mathbb{Y}}_c$

The processing in stage 3 is shown in algorithm 3 below.

Algorithm 3 OCluDAL Step 2: SVM Classification and BvSB uncertainty sampling

Input: $\mathbb{U}, \mathbb{L}$ ▷ Unlabelled, labelled datasets
Output: $\tilde{\mathbb{Y}}_c, F_\Theta$ ▷ Classified data, Fitted SVM model
$i = 1$ ▷ Iteration
while $i < I$ or stopping criterion not satisfied **do** $F_\Theta \leftarrow$ TrainClassifier$(\mathbb{L})$
$P_\mathbb{U} \leftarrow F_\Theta(\mathbb{U})$ ▷ Probability distribution
$\mathbb{S}_u \leftarrow$ UncertaintySampling$(P_\mathbb{U})$ ▷ Sample uncertainty dataset
$\mathbb{L}_u \leftarrow$ Annotation$(\mathbb{S}_u)$ ▷ Labelled uncertainty dataset
$\mathbb{L} \leftarrow \mathbb{L} \cup \mathbb{L}_u$ ▷ Updated labelled dataset
$\mathbb{U} \leftarrow \mathbb{U} \backslash \mathbb{L}_u$ ▷ Updated unlabelled dataset
$t \leftarrow t + 1$
end while
$F_\Theta \leftarrow$ TrainSVM$(\mathbb{L})$
$\tilde{\mathbb{Y}}_c \leftarrow F_\Theta(\mathbb{U})$

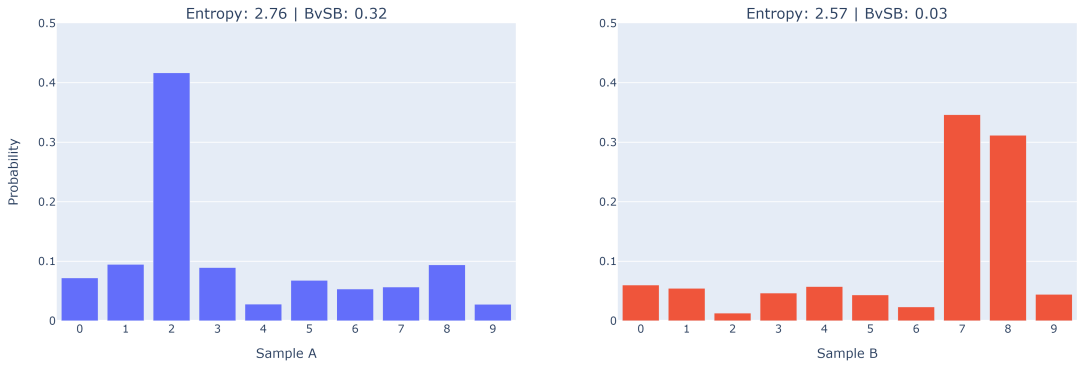

FIGURE 3.7: Comparison of Uncertainty Sampling Methods: BvSB and Entropy

Using Entropy as a scoring metric, sample A would be chosen for annotation as the higher EP value indicates a higher uncertainty. However, the high uncertainty value is caused by the relatively high probabilities of classes 1, 3 and 8. The classifier is not unsure of the prediction as activity 2. Whereas for sample B, the classifier is unsure between 7 and 8, but the remainder of probabilities are relatively much lower and so a lower entropy score is derived. Using BvSB, sample B would get chosen as the distance between the best and second best is much smaller, and so the selection would provide more information to the classifier

3.5 Stage 4: Initialise Anomaly Detection Model

Once the remainder of the unlabelled data has been classified in the previous stage, data processing for the next stage can take place. This involves grouping together continuous behaviour of the same action into bins. An example of these groupings can be seen in figure 3.8 below.

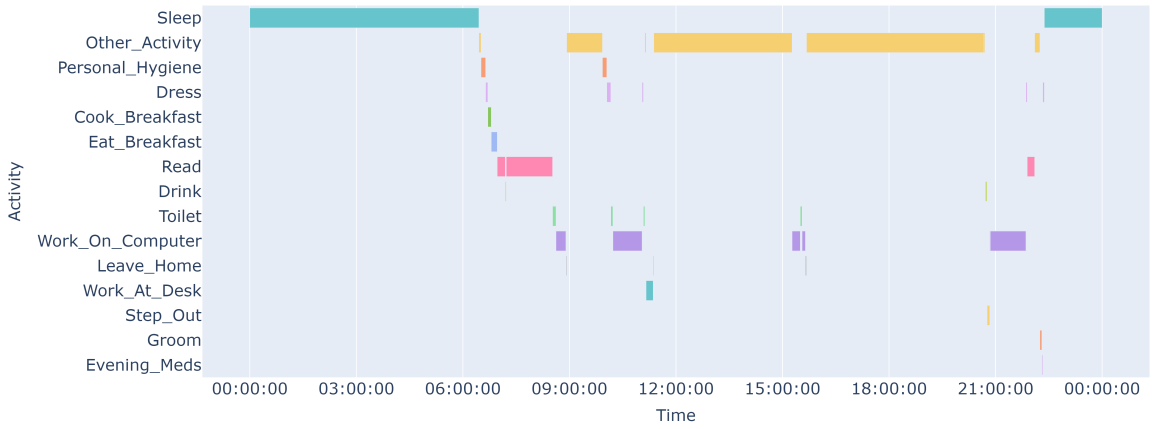

FIGURE 3.8: Activity Groupings for a Single Day from the CASAS HH111 Dataset

These activity groups are then subject to another round of feature extraction. The extracted features, first proposed in [47], are outlined in Table 3.2.

The study in [47] solely considers sleeping behaviour and patterns, disregarding the remaining data. Instead of discarding this data, this research proposes an alternative feature table (Table 3.3) that discretizes the other activities into individual days rather than isolated activities. This approach provides a more robust method for examining behaviour.

Again, the data is standardized as the outlier detection models in the ensemble can be sensitive to different scales of data. The data is split into training and testing data. In previous work established in [47], they identify a minimum amount of 30 days required for training as this allows the weekly and daily trends to be fully incorporated into the modelling. Using less than this results in significantly higher rates of false positives in the testing data. The training data is then divided further into k-folds. In this methodology, k is set to 3 and the overflow of data is put into the last fold.

Ensemble learning is a machine learning technique that combines multiple models to form an optimal combined model [56]. The idea is that the different parts of the ensemble have different strengths and weaknesses, which are shared or mitigated when grouped. In this paper, the ensemble consists of four different methods for outlier

TABLE 3.2: Anomaly Detection Feature Extraction for Sleep Behaviour

Name	Description
Start time	Start time of activity. This is measured in hours and minutes using a 24 hour clock.
Duration	The total time in minutes the particular activity is going on for.
Number of interruptions	An interruption is only considered when the previous activity is sleeping, and the interruption time is under 60 minutes. If the interruption time is greater than 60 minutes, the interruption is considered a separate activity.
Duration of interruptions	The sum time of the durations mentioned above.
Day of activity	A value from 0 to 6 corresponds to a particular day of the week. This is to allow for trends in weekly behaviour to occur.
Weekday or weekend	A value from 0 to 1 corresponds to either weekdays or weekends. This is to allow for trends between weekdays and weekends to occur.

TABLE 3.3: Anomaly Detection Feature Extraction for Non-sleep Behaviour

Name	Description
Day of activity	A value from 0 to 6 corresponds to a particular day of the week. This is to allow for trends in weekly behaviour to occur.
Weekday or weekend	A value from 0 to 1 corresponds to either weekdays or weekends. This is to allow for trends between weekdays and weekends to occur.
Total frequency*	An integer value that counts the occurrences of specific activity for a given day. **There is a frequency column for each of the activities in the dataset.**
Total duration*	The sum duration in minutes of all the events of a specific activity for a given day. **There is a duration column for each of the activities in the dataset.**

detection: Isolation Forest (IF), Local Outlier Factor (LOF), One-Class Support Vector Machine (OC-SVM), and Robust Covariance Estimation (RCE). OC-SVMs were detailed in an earlier section.

The first method used is Isolation Forest, which was introduced in [57]. This method uses a random forest model where each decision tree randomly selects a feature and a splitting point between the maximum and minimum values. The number of splittings required for an observation to become isolated is a measure of normality. On average, outliers will have much smaller paths and become isolated much sooner.

The second method is Local Outlier Factor (LOF) [58]. This method produces a score that reflects the abnormality of a given data point by measuring the ratio of the observation's local density with that of its k-nearest neighbours. Abnormal observations are defined as having a much lower density of points around them compared with their k-nearest neighbours. This method takes into account both local and global properties of the dataset when forming its predictions.

The third method is Robust Covariance Estimation (RCE), which involves fitting an elliptic envelope to the data and defining outliers as observations that are beyond a certain distance from the envelope. This method assumes that the data is from a known distribution.

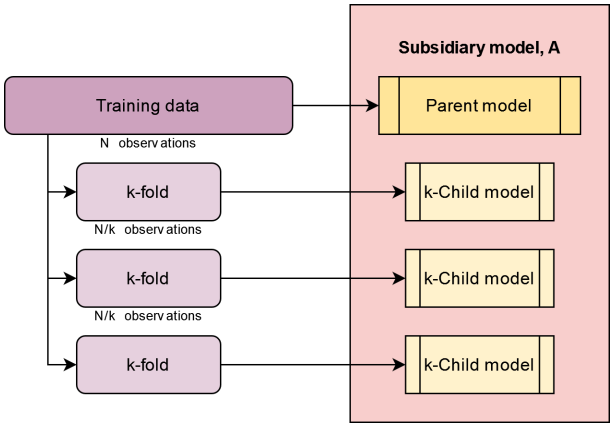

FIGURE 3.9: Structure of a Subsidiary Model, A
All of the child models and the parent model are fitted to data using the same method, only the data changes between them.

To form the ensemble, it is broken down into four smaller subsidiary ensemble models of the same structure, but with different novelty detection methods taking place under the hood. Each of these subsidiary ensembles has the structure of a singular, "strong" parent classifier and k "weak" child classifiers [59] as shown in figure 3.9. The parent classifier is trained on the entire set of training data while each of the k-child classifiers is trained on each of the k-folds of data. This approach is inspired by bagging [60].

3.6 Stage 5: Assembling and Training Ensemble

The final ensemble architecture, as introduced in [47], can be described as:

$$\text{Ensemble model, } E \equiv \{A \in M, a\} \tag{3.8}$$

where a is the list of child models m, where each one is trained on a different k-fold of the data, such that:

$$a \equiv \{m_1, m_2, ...m_k\} \tag{3.9}$$

where M is the parent model that is trained on all of the data available.

Each subsidiary model is identical in structure but trained on different algorithms: IF, LOF, OC-SVM, and RCE. A weight is associated with each of the subsidiary models that penalises or amplifies their scoring to allow the ensemble to adapt if a particular subsidiary model is not well suited for the given data. These weights are learnt during training. An illustration of the architecture can be seen in Figure 3.10. Explanations of the different domains and terminology can be found below.

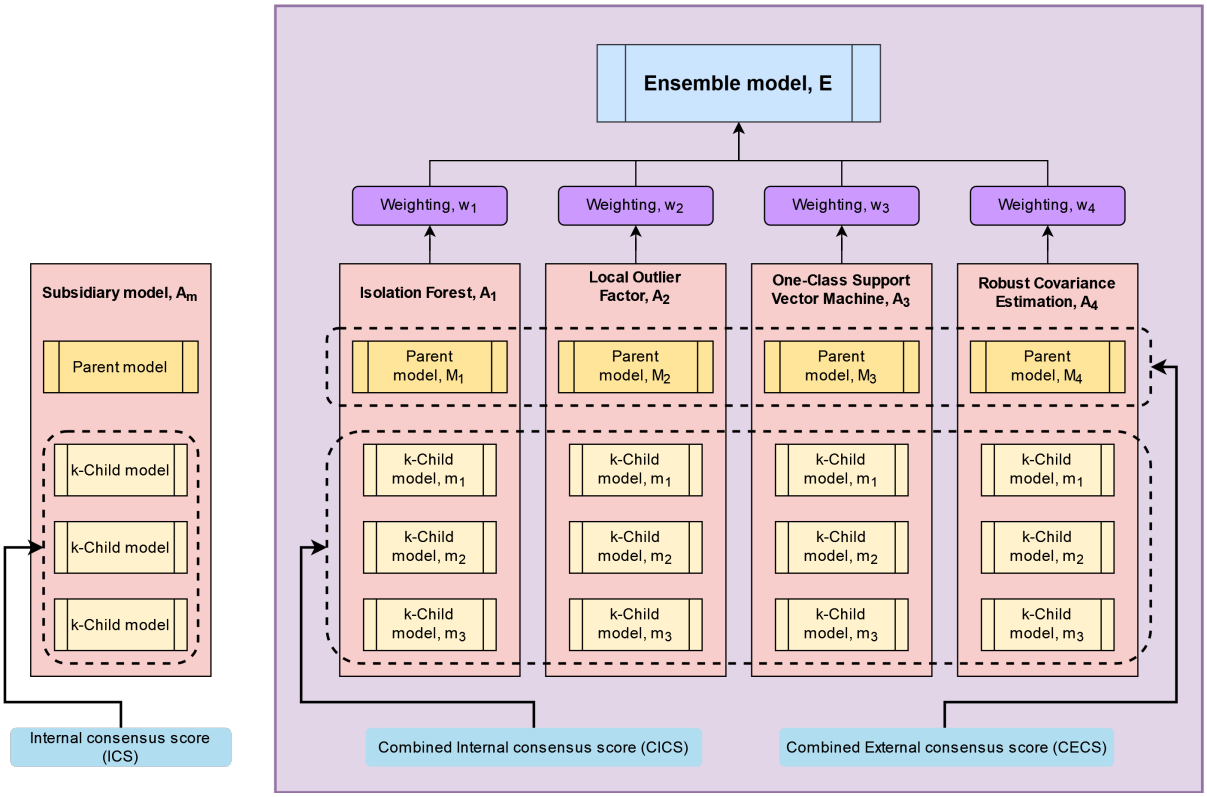

FIGURE 3.10: Ensemble Architecture Schematic

The four subsidiary models work together in a network to create a scoring system for activity observations. A scoring system, composed of the internal and external consensus scores is explained in greater detail below.

The approach relies on an internal and external scoring process to build trust among the constituent parts of the ensemble. The internal scores are computed using the child models, and the external scores by the parent models. For each observation in a set of n samples, $x = x_1, x_2, ..., x_n$ of d-dimensional data ($X \in \mathbb{R}^d$), internal and external votes on if the observation is an inlier are cast. The votes received by an observation from the subsidiary models are termed the Internal Consensus Vote (ICV). An observation is considered an inlier if it receives one or more ICVs. The Combined External Consensus Vote (CECV) is the aggregated vote for an observation being an inlier, determined by the majority rule.

Each model's weighting is a value between 0 and 1, reflecting the trust placed in the specific subsidiary model. Higher weightings signify greater importance for the model's scoring. The weightings are derived by comparing ICV and CECV for each observation. A difference between these values indicates an incorrect prediction, and the subsidiary model is penalized accordingly. The weights are initialized as 1 and penalized in proportion to the sum of errors, as shown in Equation 3.10:

$$w_f = w_i - \frac{e}{n} w_i \tag{3.10}$$

where w_f is the final weight after penalization, w_i is the initial weight, e is the number of incorrect predictions, and n is the total number of data points. Once the model has been trained and the weights adjusted, the observations can be scored. Each of the subsidiary models has a decision function component that assigns an observation a score, where the larger the number, the more normal it is. Lower values correspond to outliers or anomalies. The Internal Consensus Score (ICS) is the mean value assigned to an observation from a subsidiary model a's child models, as shown in Equation 3.11 and algorithm 4

$$ICS_{x,a} = \frac{1}{k} \sum_{j=1}^{k} v_{x,j} \tag{3.11}$$

where k is the number of child models, and $v_{x,j}$ is the vote assigned to observation x from child model j.

Algorithm 4 ICS: Compute internal scores and votes

Input: x_i $a \equiv \{m_1, m_2, ..., m_k\}$ ▷ Observation, Child model list
Output: ICV_i, ICS_i ▷ Internal consensus score, vote for x_i
  for m_k in a **do**
    $v \leftarrow \text{Vote}(m_k, x_i)$ ▷ Vote as inlier or outlier
  end for
  $t_i = \sum v$ ▷ Sum of scores
  $ICS_i = \frac{1}{k} t_i$
  if $t > -1$ **then** ▷ If any child has voted as an inlier
    $ICV_i = 1$ ▷ x_i is inlier
  else
    $ICV_i = 0$ ▷ x_i is outlier
  end if

The Combined Internal Consensus Score (CICS) is the weighted mean across the ICSs of each subsidiary, as shown in Equation 3.12.

$$CICS_x = \frac{1}{m} \sum_{i=1}^{m} ICS_{x,i} * w_i \tag{3.12}$$

where m is the number of subsidiary models (4). A demonstration of the process is shown in Algorithm 5

Algorithm 5 CICS: Compute combined internal scores

Input: $x_i, W \equiv \{w_1, w_2, ..., w_m\}, E \equiv \{A_1, A_2, ..., A_m\}$ ▷ Observation, Weights, Ensemble
Output: $ICV_i, CICS_I$ ▷ Internal consensus vote, combined score for x_i
  for A_m in E **do**
    $a \leftarrow A_m$ ▷ Child model list
    $ICV_i, ICS_i \leftarrow \text{ComputeInternal}(a)$ ▷ Algorithm 4
    $\rho \leftarrow ICS_i * w_i$ ▷ Weighted score
  end for
  $CICS \leftarrow \frac{1}{m} \sum \rho$

The Combined External Consensus Score (CECS) is the mean value assigned to an observation by the parent models across all subsidiaries, as shown in Equation 3.13:

$$CECS_x = \frac{1}{m} \sum_{i=1}^{m} V_{x,j} \tag{3.13}$$

where $V_{x,j}$ is the value of the vote assigned to observation x by parent model M_j. A demonstration of this process is shown in Algorithm 6

The normality score assigned to an observation is calculated as the mean value of the CICS and CECS, as shown in Equation 3.14.

$$N_x = \frac{CICS_x + CECS_x}{2} \tag{3.14}$$

Algorithm 6 CECS: Compute external scores and votes

Input: x_i, $E \equiv \{A_1, A_2, ..., A_m\}$ ▷ Observation, Ensemble
Output: $CECS_i, CECV_i$ ▷ Combined external consensus score, vote for x_i
  for M_j in E **do**
    $V \leftarrow \text{Vote}(M_j, x_i)$ ▷ Vote as inlier or outlier
  end for
  $T_i = \sum V$ ▷ Sum of scores
  $CECS_i = \frac{1}{m} T_i$
  if $T_i > 0$ **then** ▷ If majority vote
    $CECV_i = 1$ ▷ x_i is inlier
  else
    $CECV_i = 0$ ▷ x_i is outlier
  end if

From the normality scores, abnormal values can be identified. In the literature, it is recommended to use a threshold value of -3σ to distinguish outliers, as described in [47]. This threshold corresponds to the three-sigma rule, which is commonly used to identify outliers in normally distributed data [61]. However, this method is only appropriate if the distribution of scores follows a normal distribution. An alternative scoring system involves using robust Z-scores [62], which are based on the median and median absolute deviation (MAD) instead of the mean and standard deviation. The standard deviation (σ) and MAD values can be calculated as follows:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{3.15}$$

$$MAD = 1.4826 \cdot \text{median}(|x_i - \text{median}(x)|) \tag{3.16}$$

In Equations 3.15 and 3.16, x_i represents the individual data points, $\bar{x}$ is the mean of the data, and N is the total number of data points. The MAD is scaled by a constant factor of 1.4826 to make it a consistent estimator of the standard deviation for normally distributed data. The robust Z-scores can then be used to identify outliers in a manner that is less sensitive to outliers and more appropriate for data that does not follow a normal distribution. A visualisation of the the deviations is shown below.

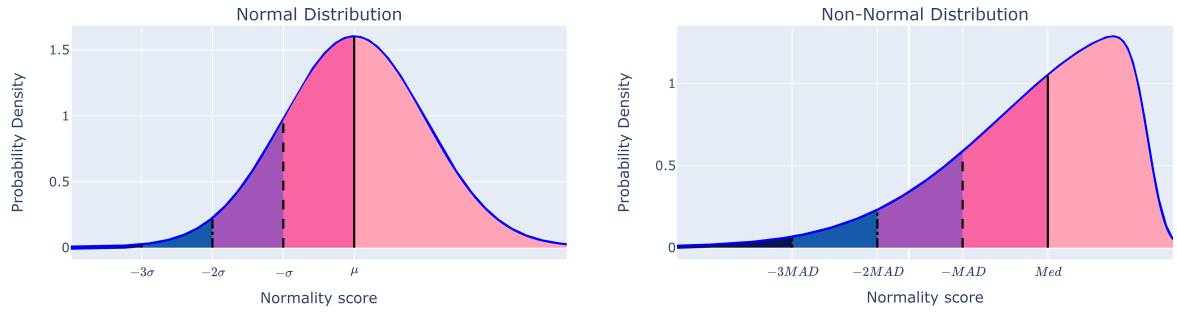

FIGURE 3.11: Normal and non-Normal distributions
If normal can be assumed, the standard deviations can be used to identify anomalies, otherwise, the robust Z-score and median absolute deviation can be used

A distribution can be determined as non-normal through qualitative inspection of the histograms or quantitatively with an absolute skew value (a measure of asymmetry) larger than 2 or an absolute kurtosis (a measure of tail heaviness) larger than 7, as described [63].

3.7 Methodology Validation

In this study, two publicly available datasets are employed to validate the proposed methodology for activity classification and anomaly detection. These datasets have been chosen due to their relevance to the study's objectives and the availability of the required data. The following is a brief introduction to the datasets used in this research:

3.7.1 Dataset: USC-HAD

The USC-HAD dataset [64] was acquired using a triaxial accelerometer and gyroscope, both mounted on the front right hip of the participants, yielding a total of six features. The dataset encompasses 12 common daily human activities, including walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping, sitting, standing, sleeping, taking an elevator up, and taking an elevator down. Fourteen participants, comprising seven males and seven females with ages ranging from 21 to 49, were part of the study. During this experimentation stage, the data from all participants and activities are combined into a single dataset.

3.7.2 Dataset: CASAS-HH111

The CASAS-HH111 dataset [65] records the activities of daily living (ADL) of a single adult volunteer residing alone in their home over a 60-day period. The documented activities include sleeping, eating, bathing, dressing, toileting, and others as shown in figure 3.8. However, the dataset does not offer any information regarding the presence or absence of abnormalities in the resident's activities.

By employing these datasets, the aim is to validate the proposed methodology and assess its potential for detecting anomalies in daily activities. The validation process and results will be presented in the Experimental Results and Discussion chapter.

3.8 Key Assumptions and Limitations

The previous sub-sections have provided details of the model's functionality, the datasets used, and the validation process. However, it is important to be aware of the following assumptions and limitations. A full evaluation of their impact on the ability of the model to accurately detect abnormal behaviour will be provided within the Results section.

1. The classification routine was developed based on the USC-HAD dataset containing six dimensions of data. As a result, the performance of the classification method may not generalize well to datasets with different dimensions or feature sets.

2. The proposed concept relies on continuous accelerometer data for a period of two months. Obtaining such long-term and continuous data can be challenging due to factors such as battery life, device maintenance, and participant adherence.

3. The level of detail in the activity descriptions that can be classified is not consistent between the two datasets used for the models. For example, the classification model handles activity classification of simple movements such as walking, sitting and turning. For the anomaly detection model, the activity classes are more complex, dealing with entertaining guests, going to the toilet, and making breakfast. This inconsistency may lead to misinterpretations or misclassifications of activities when applying the methods to new datasets. Further development of the classification model may be required to encapsulate the more complex activity types.

4. The original CNDE approach primarily focused on sleep activities and did not fully address daytime actions. To make the methodology valid and applicable to daytime activities, additional steps or modifications were required. In this study, an alternative feature table, which is an original invention, has been introduced as a potential solution and adaptation to better address daytime activities. Although this new feature table has not been validated through a published research paper, it demonstrates an effort to improve the methodology's applicability to a broader range of activities.

5. A major assumption for creating a baseline model is that the users' general activity does not deviate from that of the training data. If the users' behaviour changes over time from the point of training, all of the predictions may become irrelevant or incorrect.

6. Some of the anomaly detection models utilized in this study rely on specific data distributions to produce informative samples. If the actual data distributions differ significantly from the assumptions made by the models, the performance of the anomaly detection methods may be compromised.

7. The CASAS-HH111 dataset used in this study does not provide known anomalous events to search for. Consequently, the evaluation of the methods is limited to the investigation of false positives. False negatives, which represent undetected anomalies, cannot be quantified in this study.

4. Experimental Results and Discussion

4.1 Introduction

This chapter utilizes publicly available datasets to validate the methodology outlined in Chapter 3. For the anomaly prediction model to generate informative results, the training data must span at least 30 days. Utilizing shorter training data periods may lead to a high number of false positives during the testing data evaluation. This is attributed to the increased likelihood of classifying unseen observations in the training data as errors when they appear in the testing data. Although publicly available accelerometer data suitable for the classification model exists, it only covers a timespan of a few hours. On the other hand, two-month-long activity data is available, but it lacks the raw accelerometer data required for the initial classification stage. Consequently, this chapter is divided into two sections: one for validating the classification model and another for the anomaly detection model. This division allows for a conceptual demonstration of the process at each stage. The subsequent year's work will focus on gathering firsthand data to validate the entire process from start to finish.

4.1.1 Classification Model Validation Overview

The primary evaluation criterion for assessing the classification model's performance is the accuracy scores obtained for a specific number of annotations. The annotation process is generally resource-intensive, as it requires annotators to review a video of the user performing designated tasks and record the corresponding activity alongside the data collected from the user's wearable device. Conversely, lower final accuracy scores make the input data more susceptible to errors in the anomaly detection model. A threshold value of 95% can be utilized, with data post-processing techniques applied to reduce the output noise. The final evaluation involves comparing different models to determine which one reaches this threshold most efficiently.

4.1.2 Anomaly Detection Model Validation

The primary evaluation criterion for the anomaly detection model is the rate of false positives present in the final model. This is because the available data does not provide any information about anomalies present in the dataset. Consequently, the evaluation can only be conducted based on the detected anomalies. Each detected anomaly will be qualitatively assessed using the features outlined in Table 3.2 to determine whether they are false positives or true positives. The resulting score of the model will be calculated using the following equation:

$$FPR = \frac{FP}{FP + TP} \tag{4.1}$$

Where FPR represents the false positive rate, FP denotes the number of false positives, and TP signifies the number of true positives.

4.2 Classification Model Experimentation

This section investigates the various components and stages necessary to achieve high accuracies with minimal annotations for the classification model. Numerous algorithms involved in this process have several hyperparameters that require tuning to ensure optimal performance. As a result, the classification steps can be divided into three distinct stages, with specific optimizations applied to each, as illustrated in Figure 4.1 below.

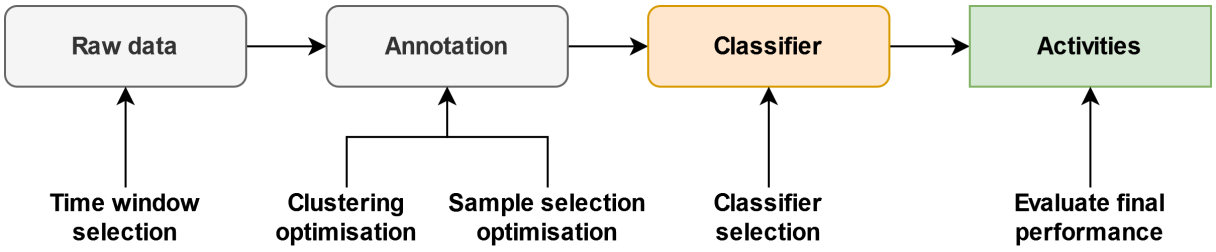

FIGURE 4.1: Overview of Classification Tuning
The location of each of the following subsections can be seen

The accuracies mentioned in the following sections are measured rates of correct predictions from the classifier.

4.2.1 Time Window Selection

In [48], the authors recommend using time windows ranging between 4 and 10 seconds. To validate this suggestion and determine the optimal parameter for the study, a sensitivity analysis was conducted, examining various time windows. A ten-fold cross-validation approach, as described in [66], was employed for this purpose. This process entails dividing the dataset into ten random, non-overlapping, and all-inclusive subsets, and then evaluating the accuracy of the model after training on each subset. The outcomes of the sensitivity analysis, along with the duration of each iteration, are displayed in Figure 4.2:

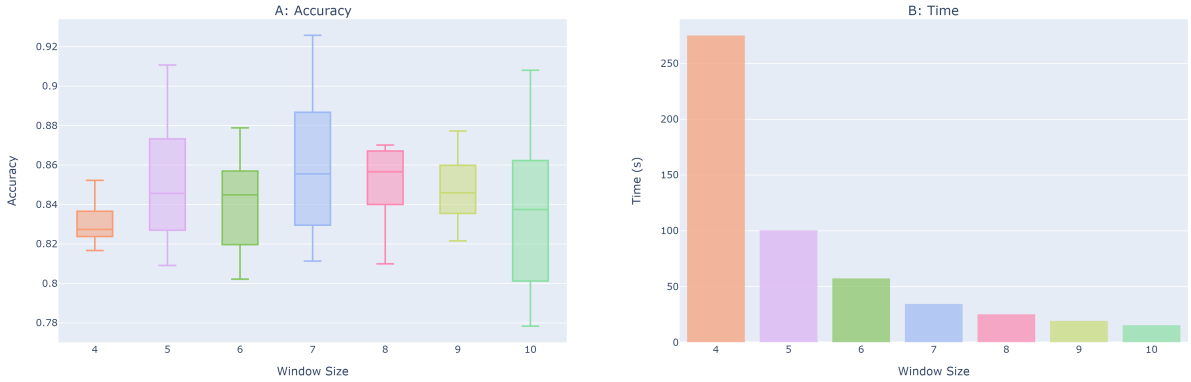

FIGURE 4.2: Varying Window Size
A comparison of accuracies (A) and processing time (B) against windowing size.

The results demonstrate that the optimal time windows are 5 and 7 seconds, both yielding very high accuracies. A 7-second time window can be preferred over a 5-second window due to the significantly reduced computation time.

These findings can be rationalized by considering that larger time windows might cause subtle distinctions in actions or activities to be obscured during feature extraction, resulting in more generalized data. Moreover, longer time windows accommodate a broader range of possible actions, leading to increased variability. Although it may be expected that smaller time windows would generate more accurate results due to the higher precision of the data, it also makes the data more vulnerable to noise and an inability to fully capture actions that require a longer duration than the window size. A mid-length value of 7 seconds appears most suitable.

The decrease in time complexity can be ascribed to the relationship between the dataset size after segmentation and the time window duration (N/w). As the time window (w) approaches zero, the dataset size tends exponentially toward infinity.

4.2.2 Sampling Selection

Entropy and BvSB sampling was introduced in section 3.4. To evaluate the different sampling methods, a performance comparison was calculated for Random, Entropy-based and BvSB-based sampling. To initiate the sampling process, 10 random samples were selected from the standing and walking forward classes. The efficacy of the sampling methods was then validated by iteratively increasing the new samples for annotation and the new accuracies calculated. The accuracies of the resulting models and the number of discovered activity classes were compared and visualized in Figure 4.3.

The results indicate that random sampling initially outperforms both entropy-based and BvSB sampling until 250 samples have been selected. This could be attributed to the algorithm confidently misclassifying an entire activity, such as walking left and walking right, as the same class. As a result, the unknown class remained undiscovered, leading to a high misclassification rate. However, once all activity classes were discovered, as shown in plot B, the accuracy of the BvSB and entropy-based sampling methods quickly surpassed that of random sampling.

Random sampling is able to discover all classes relatively quickly since there is no bias against discovering them. These results emphasize the importance of using an initial clustering routine, as it can efficiently discover all available activity classes, as demonstrated in the sections below.

4.2.3 Clustering Optimisation

As mentioned in Section 3.3, the performance of AP clustering depends on two main hyperparameters: preference and damping. However, using only the silhouette score for evaluating the algorithm's performance might not be

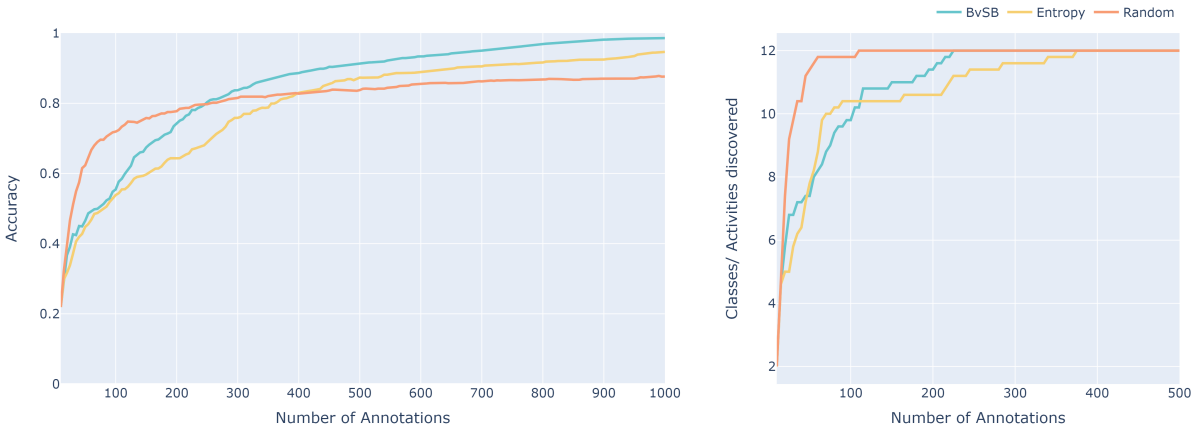

FIGURE 4.3: Comparison of Sampling Methods
Plot A illustrates the accuracy of the three sampling methods (entropy-based, BvSB, and random) as the number of selected samples increases. Plot B shows the number of discovered activity classes for each method. These results demonstrate the effectiveness of BvSB sampling in comparison to entropy-based and random sampling, particularly after all activity classes have been discovered.

adequate as the silhouette score tends to be higher when more clusters are identified. This leads in turn to much more annotations being made. It is the total sample efficiency that is the primary evaluation measure. To provide a more comprehensive assessment of the clustering performance, an adjusted score is proposed, which accounts for both the quality of the clusters and the number of samples annotated in each iteration. This adjusted score is calculated by dividing the silhouette score by the sample size. Both scores are shown in Figure 4.4.

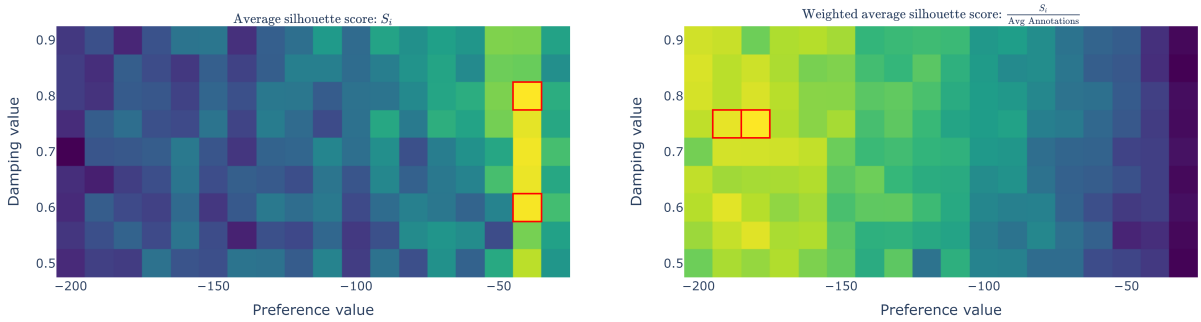

FIGURE 4.4: Comparison of the Silhouette Scores
The adjusted score provides a more robust measure of the sample efficiency as well as the clustering performance

The graphs reveal that the inclusion of annotation size significantly distorts the graph, favouring values with a lower preference. The two top-performing options for each set of scores are highlighted with red borders. This proposed analysis extends the approach suggested in [42], and as such, additional validation tests are required to evaluate the final performance when the process proceeds with the sampling selection.

To accomplish this, the various AP configurations were executed for a single iteration, and uncertainty sampling was performed until 600 samples had been reached. The accuracies of these samples could then be compared. The visualization of this process is illustrated in Figure 4.5.

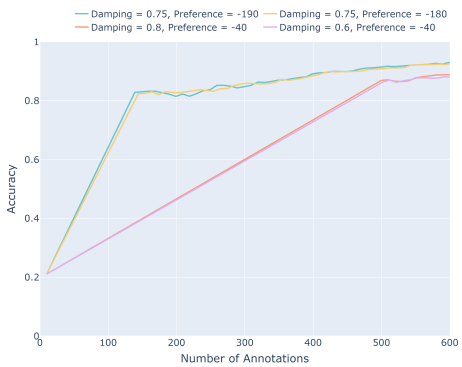

FIGURE 4.5: Validation of the Selected Samples

The plot of silhouette scores demonstrates that a higher preference value leads to higher overall accuracy for a single iteration of AP clustering. The damping value does not seem to have an influential effect. However, achieving this higher accuracy requires the annotation of many more samples, resulting in a low gradient line. This observation highlights the importance of

considering sample efficiency when evaluating clustering performance, and in turn, validates the weighted scoring method proposed earlier. A preference value of -190 and damping value of 0.75 are selected for further steps.

To validate the inclusion of clustering in the sampling process and to determine the optimal number of AP iterations. Tests were conducted to compare the effects of no clustering routines and with 1,2 and 3 iterations. This is illustrated in Figure 4.6

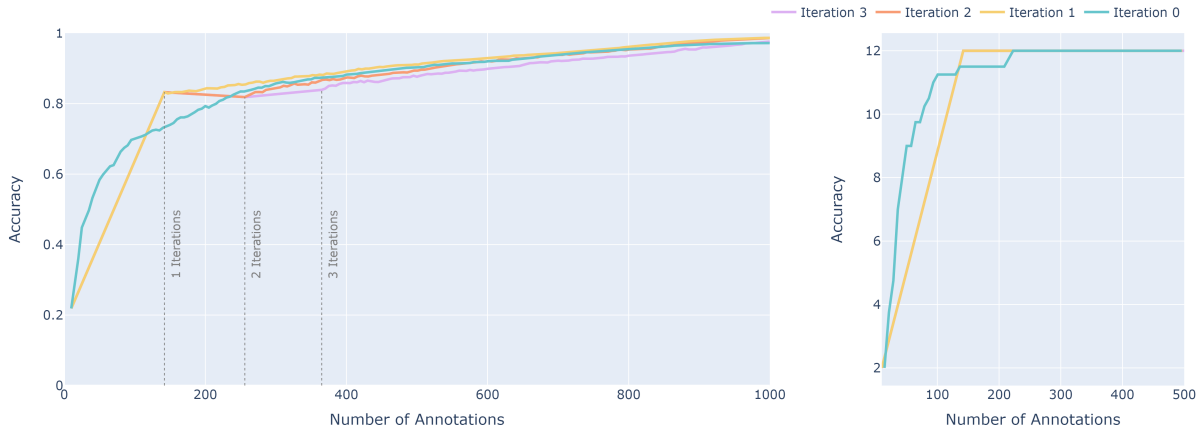

FIGURE 4.6: Impact of Clustering Iterations on Accuracy and Discovered Activity Classes
The figures illustrate the relationship between the number of AP iterations and the resulting accuracy and the number of discovered activity classes.

The first AP iteration identifies all dataset classes, improving accuracy over uncertainty sampling alone. However, the second iteration decreases accuracy, potentially due to selecting peripheral data points as new classes, introducing noise and affecting predictions with low sample numbers. As annotations increase, all methods converge to similar accuracy levels. The results suggest that using one AP clustering iteration enhances performance for smaller sample sizes.

4.2.4 Classifier Selection

In the study cited in [42], the recommended underlying model for this problem is an SVM with an RBF kernel. To validate this recommendation and explore the alternative classifiers discussed in section 3.4, cross-validation was employed in a manner similar to the time window selection process. The classifiers selected for comparison included an SVM with an RBF kernel, an SVM with a linear kernel, and KNN classifiers with k-values of 2, 5, and 10. The results are visualised in Figure 4.7.

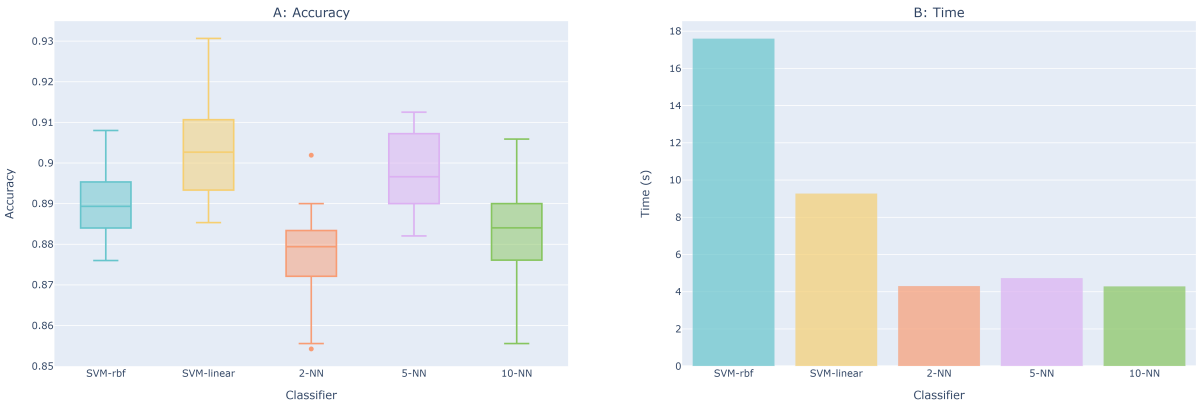

FIGURE 4.7: k-Fold Cross-Validation on Various Classifier Options

Interestingly, the linear kernel SVM outperformed the RBF kernel SVM, even with little to no tweaking of hyperparameters. The KNN classifiers also performed well, but not as well as the linear SVM. One possible explanation for the superior performance of the linear kernel SVM is the high number of dimensions in the data, which could make the data linearly separable in the high-dimensional space. Additionally, the risk of overfitting increases with the number of dimensions, and capturing noise in the data can lead to poor model generalization.

Another potential reason for the underperformance of the RBF kernel SVM is its sensitivity to data scaling. However, no errors in the data scaling process could be identified in this case. Thus, based on these results, the linear kernel SVM was chosen as the most suitable model for this problem.

4.2.5 Final Classification Model Peformance

The final classification model was assessed alongside various alternative model variations, which included the different experimental steps discussed in the preceding sections. For each method, five iterations were executed, and the average values were computed for every set of results. The properties of these model variations are summarized in Table 4.1 below. This section aims to not only validate the previous steps but also to demonstrate the achieved performance of the classification model under different configurations.

TABLE 4.1: Final Classification Comparison

Name	Sampling type	Clustering iterations	Classifier type
Final Model	BvSB	1	Linear-SVM
Model B	BvSB	2	Linear-SVM
Model C	BvSB	1	RBF-SVM
Model D	BvSB	1	5-NN
Model E	BvSB	0	Linear-SVM
Model F	Random	0	Linear-SVM

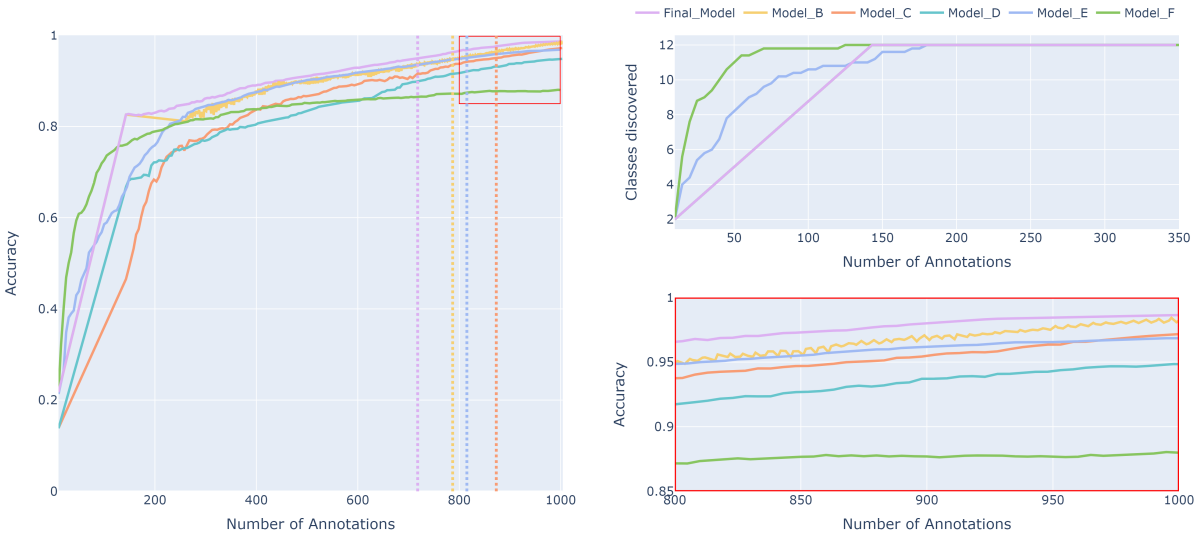

FIGURE 4.8: Comparison of Model Performance on USC-HAD Dataset
The vertical lines represent the point at which the classification accuracy exceeds a value of 95%

The final model demonstrated superior performance compared to the other variations, validating the optimization efforts applied to each step. The final model reached a 95% accuracy after only 718 samples.

Interestingly, random sampling performed remarkably well for fewer than 125 annotations, owing to its capacity to discover new activities. This can be attributed to the AP clustering selecting 133 samples in a single instance for annotation. In that single iteration, it outperformed random sampling and identified all of the classes.

For scenarios with extremely limited samples, employing random annotation might prove to be the most effective approach.

4.3 Anomaly Detection Experimentation

This section validates the ensemble anomaly detection model by analyzing anomaly scores from subsidiary models and their combined distribution. Specific anomalous events are further examined for detailed analysis. The experimentation is divided into two parts: sleep data and remaining activities, allowing deeper understanding of

the model's performance in different contexts. Training data consists of the first 30 days, while the second 30 days serve as testing data. In the sleep data, observations refer to single sleep sessions, with a 60-minute interval separating sessions. In general activity data, an observation refers to an entire day. The primary evaluation criterion is the false positive rate, assessing the ensemble model's accuracy in identifying anomalies without excessive false alarms, ensuring practicality and reliability for real-world applications in ambient assisted living environments.

4.3.1 Anomalous Sleep Analysis

Scoring Distributions

Since each subsidiary model exhibits unique behaviour in response to the dataset, it is crucial to examine their respective scoring patterns and ensure they perform as expected. Figure 4.9 displays the distributions of the models' scores.

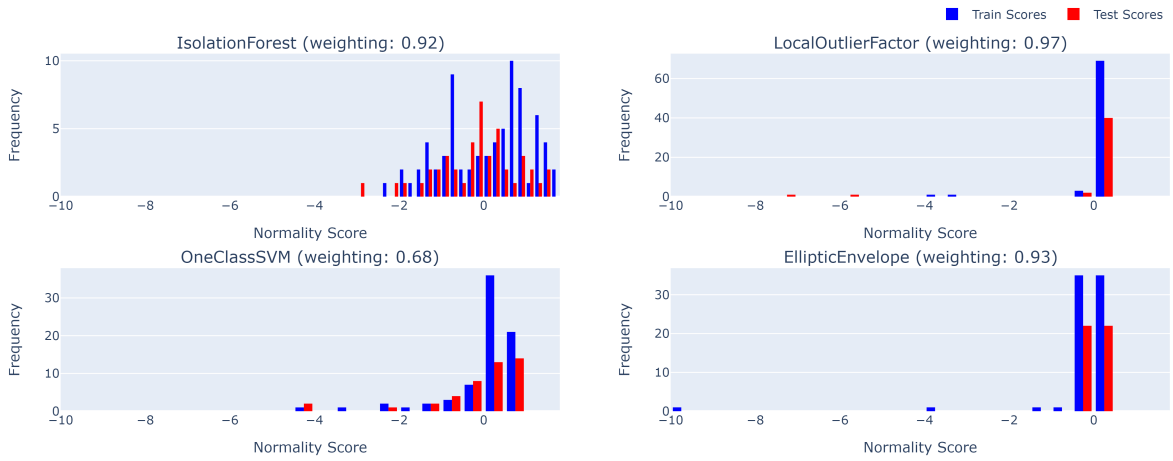

FIGURE 4.9: Sleep Scoring Distributions of Subsidiary Models

The distinct distributions in these figures confirm the benefits of employing an ensemble approach, as each model can potentially detect new anomalies or counteract undesired scores originating from a biased model. Figure 4.10 presents the combined distribution, where weightings are applied based on each model's overall performance.

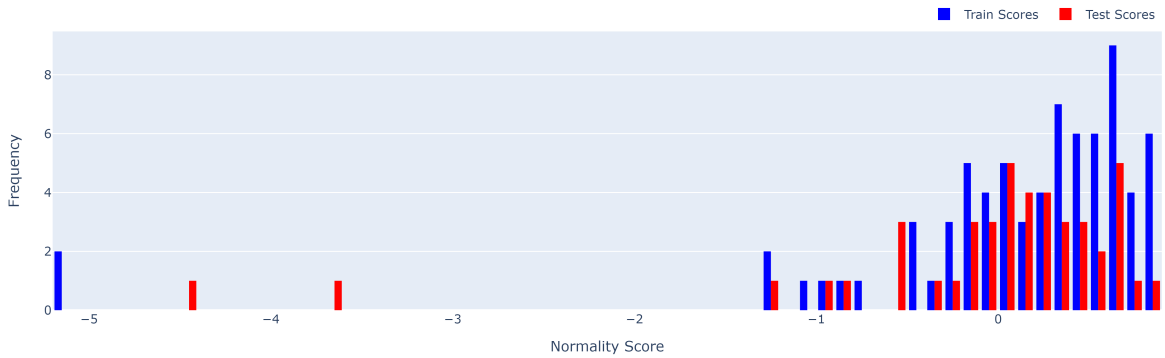

FIGURE 4.10: Combined Weighted Distribution of Sleep Normality Scores

The figure reveals a trend that deviates from a normal distribution, with a noticeable rightward skew. The measured skew value of -3.79 and kurtosis value of 16.24 both surpass the threshold values for assuming normal distributions. As a result, a robust Z-score is selected as the method for determining the anomaly threshold.

Model performance

The normality scores are plotted against the data, including median, $-MAD$, $-2MAD$, and $-3MAD$ values in the figure. Circled values represent anomalous results identified by the model. The entire algorithm is run multiple times with randomized k-folds, and the scores are averaged. Figure 4.11 illustrates the relative scoring of each observation and the anomalous results are explained in more detail in Table 4.2.

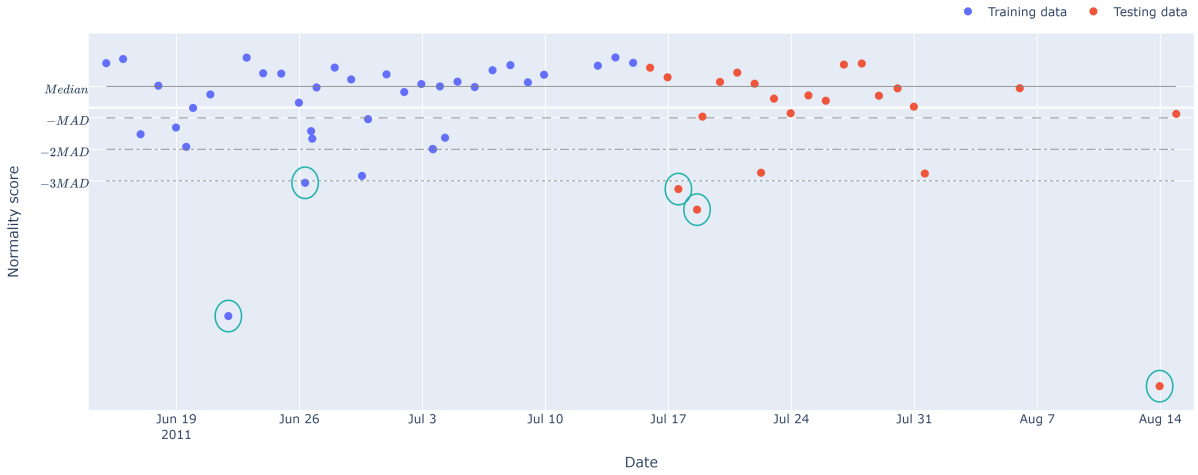

FIGURE 4.11: Normality Scores Plotted Against Data Sleep Activities

TABLE 4.2: Anomalous Sleep Activity

Anomalous events detected by the model with corresponding feature data and reasons for classification. False positives are highlighted in red and true positives are highlighted in green

Start time	Duration (minutes)	Number of interruptions	Duration of interruptions	Day of activity	Reason
21/06 22:55	481.03	3	53.83	Tuesday	Numerous lengthy interruptions
26/06 07:47	2.45	0	0	Sunday	This observation is a false positive due to an error in the data processing. It is likely due to a bed sensor going off when it was not supposed to.
18/07 15:09	54.4	0	0	Monday	Unusual for a Monday afternoon. There no naps present in training set.
22/07 06:36	30.87	0	0	Friday	Disruption at 5:30am not classified as interruption, but going back to bed afterwards was unusual.
13/08 23:04	510.08	8	26	Saturday	Numerous lengthy interruptions.

Among the detected anomalies, four can be considered genuine anomalous occurrences. This results in a false positive rate of 20%. One issue that becomes evident is the presence of anomalous results within the training data. For the model to be unbiased, it must contain very few anomalies. To address this issue, one can filter known abnormal behaviours, such as frequent sleep interruptions and late bedtimes, and remove them from the training set, reducing the bias towards ignoring such events. Conversely, actions not present during training may also lead the model to classify regular activities as anomalous if they have not been previously encountered. A potential solution to this problem is to collect data for longer periods, increasing the model's exposure to a broader range of activities. Another challenge arises from noise in the dataset, which can cause the model to overfit the data. This issue can be mitigated by implementing additional post-processing steps between the classification and anomaly detection stages.

4.3.2 Anomalous Activity Analysis

Scoring Distributions

A notable distinction between the sleep and activity datasets is the increased variation in the latter. The activity dataset consists of 54 features, as opposed to the 6 features for sleep, with a frequency and duration variable for each activity present. The calculated weightings of the subsidiary models is:

TABLE 4.3: Subsidiary Model Weightings

	Isolation Forest	Local Outlier Factor	One-Class SVM	Elliptic Envelope
Weighting	0.77	0.97	0.13	1

For this dataset, OC-SVM exhibits poorer performance compared to the other models, and its influence on the final scoring algorithm will be minimal. This may be attributed to the high dimensionality of this dataset. Since the number of dimensions exceeds the number of observations or days, the OC-SVM likely struggled to establish an appropriate decision boundary. The combined weighted distribution is presented in Figure 4.12.

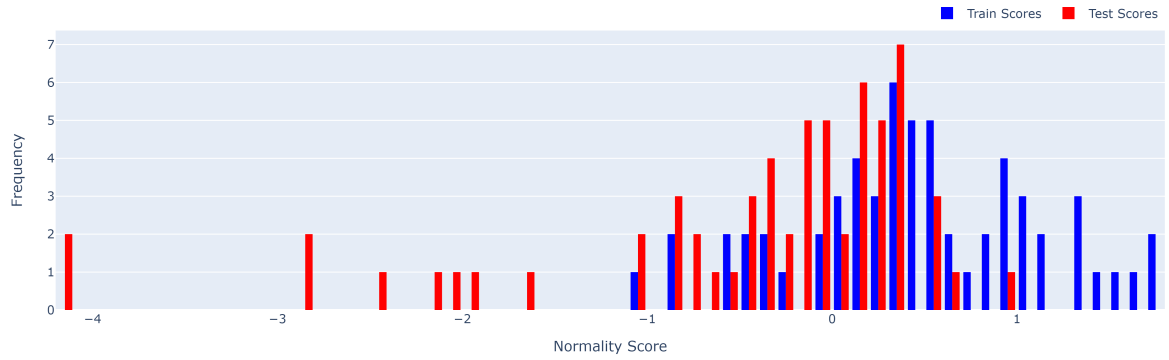

FIGURE 4.12: Combined Weighted Distribution of Activity Normality Scores
Despite a few erroneous observations to the left of the main data body, both skewness and kurtosis fall within the limits for normality, allowing for the assumption of a normal distribution.

The measured skew value of -1.87 and kurtosis value of 5.13 are both below the threshold values required for assuming normal distributions. Consequently, the distribution can be considered normal, allowing the mean and standard deviation to be employed in determining the mean.

Model Performance

The normality scores are plotted against the data, including μ, $-\sigma$, -2σ, and -3σ values in the plot. Circled values represent anomalous results identified by the model. The entire algorithm is run multiple times with randomized k-folds, and the scores are averaged. The results are illustrated in Figure 4.13.

One immediate observation from the plot is the presence of a periodic trend that can be associated with the days of the week. This may suggest that the weekday and weekend features have not been effectively incorporated into the prediction process. In future models, this issue could be addressed by developing individual models for each day of the week and ensuring the analysis of trends on specific days. However, given the limited data available, the current dataset would not provide enough days of data to conduct a meaningful analysis. In contrast to the sleep data, there appear to be no anomalous results present in the dataset. To gain a better understanding of the dataset and identify normal behaviour, the average frequency and duration of participants' daily activities are visualised in Figure 4.14. This allows a greater contextual understanding of the participant's routine during training.

The anomalous results can now be individually investigated to determine whether they are true or false positives. For each of the following days, a plot displays the exact difference in frequency and percentage difference in duration for each activity, comparing that specific day to the average values calculated across the entire dataset. The checks take place from the most abnormal to the points just over the threshold.

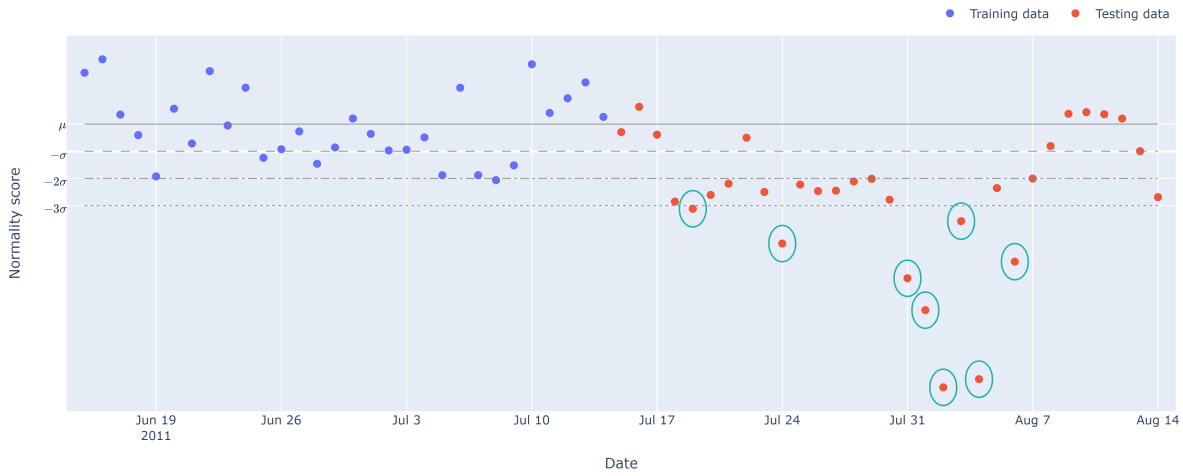

FIGURE 4.13: Normality Scores Plotted Against General Activities

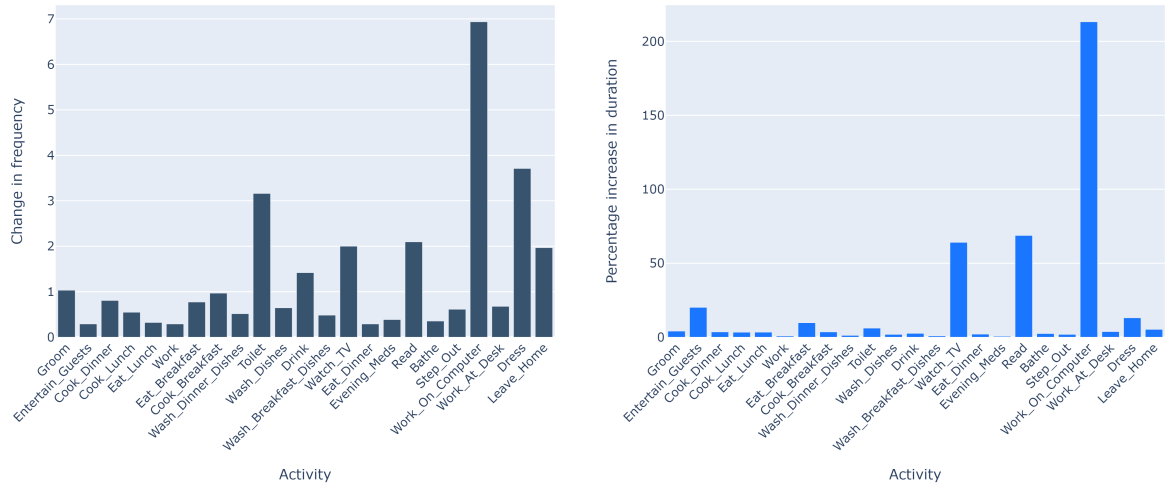

FIGURE 4.14: Average Daily Activity

Day: 2011-08-02

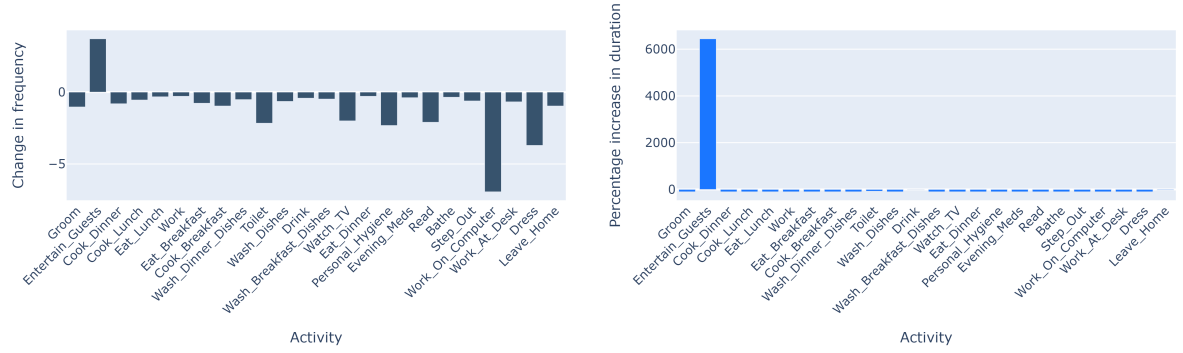

FIGURE 4.15: Abnormal Day: 2011-08-02

As evident from the plot, the abnormal score can be attributed to the significant increase in the activity 'Entertain Guests', along with a decrease in other activities. All anomalous days from 31/07 to 06/08 exhibit similar plots, and the reason for flagging them is the same. During this week, it is likely that the user had guests staying with them. These data points can be considered anomalous behaviour since, for a user with neurodiverse conditions, such a drastic change in social dynamics could be perceived as draining or intensive. This situation would warrant

a caregiver to visit and check on the individual's well-being both during and after the visit.

Day: 2011-07-24

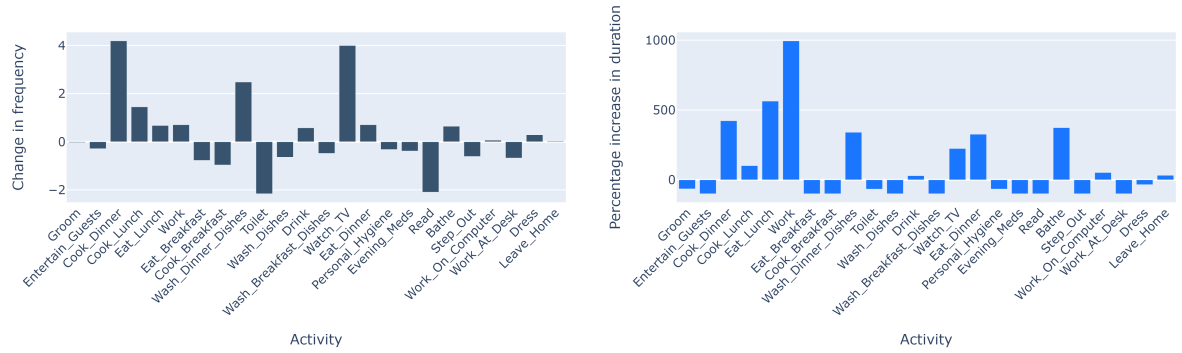

FIGURE 4.16: Abnormal Day: 2011-07-24

The anomalous result can likely be attributed to the increase in work. This could be linked to an increase in job or study-related stress, which would warrant a caregiver to visit and check on the user. As a result, this day could be considered a true positive.

Day: 2011-07-19

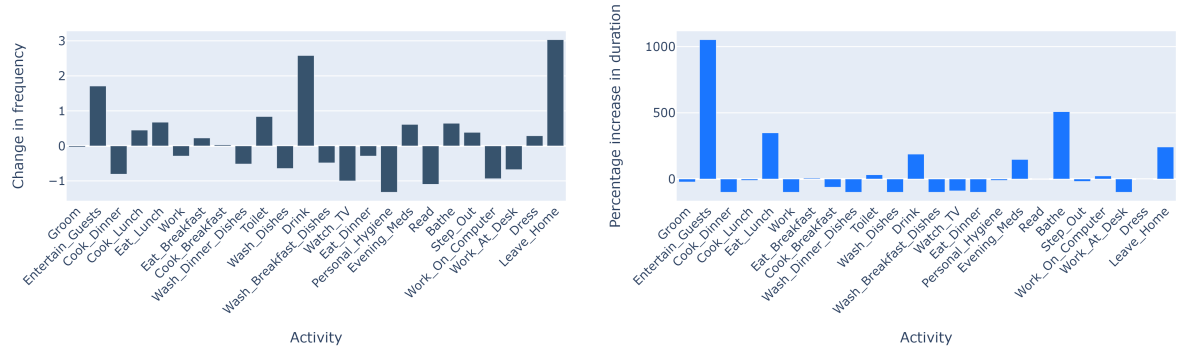

FIGURE 4.17: Abnormal Day: 2011-07-19

Again, there has been an increase in the duration of entertaining guests. However, for this day, the increase is not particularly large and could be considered a false alarm. There is very little noticeable change in activities other than that.

4.3.3 Final Anomaly Model Discussion

Out of the 8 anomalous days detected, 7 can be considered as true anomalous days, resulting in a false positive rate of just 12.5% for the general activity data. This demonstrates that the model performs well in identifying days with unusual behaviour.

The single day that could be described as incorrectly assigned an anomaly sits just below the threshold value. In cases where the model consistently flags days that may be slightly unusual but not significant enough to be classified as anomalies, the threshold value for assigning anomalies can be adjusted. By fine-tuning the threshold value, the model can be made more or less lenient, depending on the desired level of sensitivity.

It is important to note that the model's performance may improve with more extensive data, which can provide a better representation of the user's typical behaviour. Further analysis of individual days of the week or incorporation of additional contextual factors may also enhance the model's ability to detect true anomalies.

5. Conclusions and Future Work

The methodology presented in this report outlines a multi-stage process aimed at conceptually demonstrating anomaly detection using data gathered from a user's wearable device. The approach demonstrated promising results in activity classification and anomaly detection, providing a solid foundation for future research

The first half of the process focuses on activity classification, which entails determining the user's activities based solely on the accelerometer data obtained from wrist or hip-worn devices. Optimization of this stage is crucial to reduce the workload of annotators responsible for recording an extensive amount of labels. The final classification process employs a linear-based SVM to classify activities on 7-second windowed data, utilizing a single round of affinity propagation and BvSB uncertainty sampling to iteratively select informative samples for annotation. This method achieves a 95% accuracy in 718 samples, significantly surpassing additional methods suggested and demonstrating a 10% higher accuracy than randomly selecting samples. High accuracies in this stage justify the assumption that the incoming data for the anomaly detection model is reliable and not error-prone.

The second half of the process emphasizes detecting anomalies from the classified data, involving both a sleep model and a general activity model. The false positive rates for determining anomalous days are 12.5% out of 8 days and 20% out of 5 days for anomalous sleep behaviour, indicating a successful identification of anomalies in the data without excessive sensitivity.

However, several limitations need to be addressed in future work:

1. Transitioning from a conceptual design to a more robust system necessitates the use of a consistent data source throughout the entire process. This requires collecting accelerometer data from a single user for at least two months.

2. The complexity of the activities discovered is another limitation. The classification dataset used includes simple tasks such as walking and standing, while the anomaly dataset contains tasks like cooking and entertaining guests. The classification model may need additional adaptations to identify these tasks, such as incorporating new data sources like door sensors to indicate rooms or physiological data to determine the user's mood in an attempt to classify more complicated tasks.

3. Ensuring that the training data for the anomaly model is devoid of anomalies is crucial, as their inclusion may lead to skewed data and misclassification of anomalies as normal. Filtering the data of known anomalies in future models can mitigate this issue.

4. A significant limitation of the current process is the evaluation of false negatives, which are anomalous observations misclassified as normal. Future editions could address this by synthesizing false datasets and including artificial anomalies, such as falls, erratic sleep behaviour, excessive toilet trips, or a general lack of activity. Knowing the number of anomalies before prediction allows for a more robust accuracy calculation, evaluating both false positive and false negative rates.

5. Improving the generalizability of the model is important. The current model is based on specific datasets and may not perform well when applied to other datasets or users with different activities and patterns. In future work, additional datasets can be incorporated, and the model can be adapted to accommodate diverse user activities, patterns, and demographics.

By addressing these limitations and further refining the approach, a more accurate and reliable system for detecting anomalies in the activities and behaviours of residents in supported living situations can be developed. This could ultimately contribute to enhancing their quality of life and independence, while providing timely information for caregivers to intervene when necessary, promoting better health outcomes and more efficient allocation of care resources.

References

[1] *Key facts and figures about adult social care*. July 2021. URL: https://www.kingsfund.org.uk/audio-video/key-facts-%20figures-adult-social-care.

[2] *Summary: The State of Ageing 2022*. URL: https://ageing-better.org.uk/summary-state-ageing-2022.

[3] *Research and statistics*. Sept. 2022. URL: https://www.mencap.org.uk/learning-disability-explained/research-and-statistics.

[4] National Institute for Health CKS and Care Excellence. *How common is it?* 2020. URL: https://cks.nice.org.uk/topics/autism-in-adults/background-information/prevalence/.

[5] Coral Living. *Home*. Oct. 2020. URL: https://coral-living.co.uk/.

[6] A Hasan Sapci and H Aylin Sapci. "Innovative Assisted Living Tools, Remote Monitoring Technologies, Artificial Intelligence-Driven Solutions, and Robotic Systems for Aging Societies: Systematic Review". en. In: *JMIR Aging* 2.2 (Nov. 2019), e15429.

[7] Amazon. *Alexa: Smart Home*. 2023. URL: https://www.amazon.co.uk/b?ie=UTF8&node=28247729031.

[8] Google. *Google Home*. 2021. URL: https://home.google.com/welcome/.

[9] Julia Offermann-van Heek, Eva-Maria Schomakers, and Martina Ziefle. "Bare necessities? How the need for care modulates the acceptance of ambient assisted living technologies". In: *International Journal of Medical Informatics* 127 (2019), pp. 147–156. ISSN: 1386-5056. DOI: https://doi.org/10.1016/j.ijmedinf.2019.04.025. URL: https://www.sciencedirect.com/science/article/pii/S1386505618311675.

[10] George Demiris et al. "Older adults' privacy considerations for vision based recognition methods of eldercare applications". In: *Technology and Health Care* 17 (2009). 1, pp. 41–48. ISSN: 1878-7401. DOI: 10.3233/THC-2009-0530. URL: https://doi.org/10.3233/THC-2009-0530.

[11] J. H. M. Bergmann and A. H. McGregor. "Body-Worn Sensor Design: What Do Patients and Clinicians Want?" In: *Annals of Biomedical Engineering* 39.9 (Sept. 2011), pp. 2299–2312. ISSN: 1573-9686. DOI: 10.1007/s10439-011-0339-9. URL: https://doi.org/10.1007/s10439-011-0339-9.

[12] Guillaume Gingras et al. "IoT Ambient Assisted Living: Scalable Analytics Architecture and Flexible Process". In: *Procedia Computer Science* 177 (2020). The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020) / The 10th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2020) / Affiliated Workshops, pp. 396–404. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.10.053. URL: https://www.sciencedirect.com/science/article/pii/S1877050920323218.

[13] Filippo Palumbo et al. "Sensor Network Infrastructure for a Home Care Monitoring System". In: *Sensors* 14.3 (2014), pp. 3833–3860. ISSN: 1424-8220. DOI: 10.3390/s140303833. URL: https://www.mdpi.com/1424-8220/14/3/3833.

[14] Alan K Bourke et al. "Embedded fall and activity monitoring for a wearable ambient assisted living solution for older adults". In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2012, pp. 248–251.

[15] Charmi Jobanputra, Jatna Bavishi, and Nishant Doshi. "Human Activity Recognition: A Survey". In: *Procedia Computer Science* 155 (2019). The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology, pp. 698–703. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.08.100. URL: https://www.sciencedirect.com/science/article/pii/S1877050919310166.

[16] Frédéric Li et al. "Comparison of Feature Learning Methods for Human Activity Recognition Using Wearable Sensors". In: *Sensors* 18.2 (2018). ISSN: 1424-8220. DOI: 10.3390/s18020679. URL: https://www.mdpi.com/1424-8220/18/2/679.

[17] Damien Bouchabou et al. "A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning". In: *Sensors* 21.18 (2021). ISSN: 1424-8220. DOI: 10.3390/s21186037. URL: https://www.mdpi.com/1424-8220/21/18/6037.

[18] Zack Zhu et al. "Human activity recognition using social media data". In: *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. 2013, pp. 1–10.

[19] Samundra Deep et al. “A Survey on Anomalous Behavior Detection for Elderly Care Using Dense-Sensing Networks”. In: *IEEE Communications Surveys Tutorials* 22.1 (2020), pp. 352–370. DOI: 10.1109/COMST.2019.2948204.

[20] Sebastian Münzner et al. “CNN-based sensor fusion techniques for multimodal human activity recognition”. In: *Proceedings of the 2017 ACM International Symposium on Wearable Computers*. 2017, pp. 158–165.

[21] Artur Jordao et al. *Human Activity Recognition Based on Wearable Sensor Data: A Standardization of the State-of-the-Art*. 2018. DOI: 10.48550/ARXIV.1806.05226. URL: https://arxiv.org/abs/1806.05226.

[22] J. R. Quinlan. “Induction of decision trees”. In: *Machine Learning* 1.1 (Mar. 1986), pp. 81–106. ISSN: 1573-0565. DOI: 10.1007/BF00116251. URL: https://doi.org/10.1007/BF00116251.

[23] Cristianini, Nello and De Bie, Tijl. *Support Vector Machines*. eng. 2005.

[24] Lawrence Rabiner and Biinghwang Juang. “An introduction to hidden Markov models”. In: *ieee assp magazine* 3.1 (1986), pp. 4–16.

[25] Leif E Peterson. “K-nearest neighbor”. In: *Scholarpedia* 4.2 (2009), p. 1883.

[26] L. Minh Dang et al. “Sensor-based and vision-based human activity recognition: A comprehensive survey”. In: *Pattern Recognition* 108 (2020), p. 107561. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107561. URL: https://www.sciencedirect.com/science/article/pii/S0031320320303642.

[27] Ivens Portugal, Paulo Alencar, and Donald Cowan. “The use of machine learning algorithms in recommender systems: A systematic review”. In: *Expert Systems with Applications* 97 (2018), pp. 205–227. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.12.020. URL: https://www.sciencedirect.com/science/article/pii/S0957417417308333.

[28] Atis Elsts and Ryan McConville. “Are Microcontrollers Ready for Deep Learning-Based Human Activity Recognition?” In: *Electronics* 10.21 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10212640. URL: https://www.mdpi.com/2079-9292/10/21/2640.

[29] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[30] Keiron O’Shea and Ryan Nash. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015).

[31] Atis Elsts et al. “Energy-efficient activity recognition framework using wearable accelerometers”. In: *Journal of Network and Computer Applications* 168 (2020), p. 102770. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2020.102770. URL: https://www.sciencedirect.com/science/article/pii/S1084804520302447.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[33] Jiuxiang Gu et al. “Recent advances in convolutional neural networks”. In: *Pattern Recognition* 77 (2018), pp. 354–377. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2017.10.013. URL: https://www.sciencedirect.com/science/article/pii/S0031320317304120.

[34] Jianbo Yang et al. “Deep convolutional neural networks on multichannel time series for human activity recognition”. In: *Twenty-fourth international joint conference on artificial intelligence*. 2015.

[35] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[36] Francisco Javier Ordóñez and Daniel Roggen. “Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition”. In: *Sensors* 16.1 (2016), p. 115.

[37] Jiuxiang Gu et al. “Recent advances in convolutional neural networks”. In: *Pattern recognition* 77 (2018), pp. 354–377.

[38] Niall Twomey et al. “A Comprehensive Study of Activity Recognition Using Accelerometers”. In: *Informatics* 5.2 (2018). ISSN: 2227-9709. DOI: 10.3390/informatics5020027. URL: https://www.mdpi.com/2227-9709/5/2/27.

[39] Sojeong Ha and Seungjin Choi. “Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors”. In: *2016 international joint conference on neural networks (IJCNN)*. IEEE. 2016, pp. 381–388.

[40] Niall Twomey et al. “Unsupervised learning of sensor topologies for improving activity recognition in smart environments”. In: *Neurocomputing* 234 (2017), pp. 93–106.

[41] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[42] Haixia Bi et al. “Human Activity Recognition Based on Dynamic Active Learning”. In: *IEEE Journal of Biomedical and Health Informatics* 25.4 (2021), pp. 922–934. DOI: 10.1109/JBHI.2020.3013403.

[43] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[44] Daniel B. Araya et al. “Collective contextual anomaly detection framework for smart buildings”. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. 2016, pp. 511–518. DOI: 10.1109/IJCNN.2016.7727242.

[45] Labiba Gillani Fahad and Syed Fahad Tahir. “Activity recognition and anomaly detection in smart homes”. In: *Neurocomputing* 423 (2021), pp. 362–372.

[46] Salisu Wada Yahaya, Ahmad Lotfi, and Mufti Mahmud. “Towards a data-driven adaptive anomaly detection system for human activity”. In: *Pattern Recognition Letters* 145 (2021), pp. 200–207.

[47] Salisu Wada Yahaya, Ahmad Lotfi, and Mufti Mahmud. “A Consensus Novelty Detection Ensemble Approach for Anomaly Detection in Activities of Daily Living”. In: *Applied Soft Computing* 83 (2019), p. 105613. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2019.105613. URL: https://www.sciencedirect.com/science/article/pii/S156849461930393X.

[48] Majid Janidarmian et al. “A Comprehensive Analysis on Wearable Acceleration Sensors in Human Activity Recognition”. In: *Sensors* 17.3 (2017). ISSN: 1424-8220. DOI: 10.3390/s17030529. URL: https://www.mdpi.com/1424-8220/17/3/529.

[49] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. *A practical guide to support vector classification*. 2003.

[50] Bernhard Schölkopf et al. “Estimating the support of a high-dimensional distribution”. In: *Neural computation* 13.7 (2001), pp. 1443–1471.

[51] Peter J Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[52] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144–152.

[53] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20 (1995), pp. 273–297.

[54] David D Lewis and Jason Catlett. “Heterogeneous uncertainty sampling for supervised learning”. In: *Machine learning proceedings 1994*. Elsevier, 1994, pp. 148–156.

[55] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. “Multi-class active learning for image classification”. In: *2009 ieee conference on computer vision and pattern recognition*. IEEE. 2009, pp. 2372–2379.

[56] Thomas G Dietterich. “Ensemble methods in machine learning”. In: *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*. Springer. 2000, pp. 1–15.

[57] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[58] Markus M Breunig et al. “LOF: identifying density-based local outliers”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[59] Shai Avidan. “Ensemble tracking”. In: *IEEE transactions on pattern analysis and machine intelligence* 29.2 (2007), pp. 261–271.

[60] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24 (1996), pp. 123–140.

[61] Friedrich Pukelsheim. “The three sigma rule”. In: *The American Statistician* 48.2 (1994), pp. 88–91.

[62] Peter J Rousseeuw and Mia Hubert. “Robust statistics for outlier detection”. In: *Wiley interdisciplinary reviews: Data mining and knowledge discovery* 1.1 (2011), pp. 73–79.

[63] Hae-Young Kim. “Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis”. In: *Restorative dentistry & endodontics* 38.1 (2013), pp. 52–54.

[64] Mi Zhang and Alexander A Sawchuk. “USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors”. In: *Proceedings of the 2012 ACM conference on ubiquitous computing*. 2012, pp. 1036–1043.

[65] Diane J Cook et al. “CASAS: A smart home in a box”. In: *Computer* 46.7 (2012), pp. 62–69.

[66] Davide Anguita et al. “The’K’in K-fold Cross Validation.” In: *ESANN*. 2012, pp. 441–446.