Human Activity Recognition/Monitoring and Anomaly Detection using Radar Sensor

Yu Chen Lee

Master of Science
School of Informatics
University of Edinburgh
2023

Abstract

In the domain of human activity monitoring or monitoring systems, sensors are essential pieces of the puzzle. This project explores the viability of radar sensors for human activity monitoring in the context of sequence of activity and quality of movement. By using a radar sensor, information of the movement or activity of an individual such as coordinates, velocity, acceleration, and timestamps could be extracted. To understand the information extracted from the raw sensory data and detect anomalies, a novel data representation is suggested by taking into account the dimension of the room and segmenting the room into different zones. Based on the time spent on different locations, different heatmaps are generated to visualize the activity sequences in a uniform time interval. Machine learning algorithms including a decision fusion model, CNN model, and Temporal Convolutional Network (TCN) model are applied to the data collected to detect anomalies based on the contexts mentioned. The Decision Fusion model, utilizing an OR gate decision layer, yields highly accurate results, achieving a 95.65% accuracy due to the adeptness of the Gaussian Mixture Model in identifying anomalous movement patterns. However, the Markov chain model demonstrates limitations in detecting anomalies within the scope of activity sequences as the model could not differentiate between different activity sequences with that are similar to each other due to the reliance on the entropy rate value to identify anomalies. The CNN model records an 85.18% accuracy. Challenges arise when the data contains identical activity sequence but abnormal movement pattern or marginally different activity sequences. To adequately take into account the temporal features of the data, the TCN model is introduced, attaining perfect accuracy on the data collected.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics committee.
Ethics application number: 574863
Date when approval was obtained: 2023-07-08

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yu Chen Lee*)

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Kianoush Nazarpour. Without his unwavering support and guidance, I would not be able to complete this meaningful work successfully. He has always been helpful in providing me insights on the field of study of this project.

Secondly, I would like to extend my heartfelt gratitude to Saber Mirzaee Bafti for his invaluable advice during weekly meetings, and continuous support throughout the journey. His expertise has been instrumental in ensuring this project's successful realization.

Lastly, I would like to sincerely thank my family and my friends for the unconditional support and care provided throughout this period.

Table of Contents

Chapter 1

Introduction

1.1 Motivation and Problem Statement

In the fields of healthcare monitoring and smart home technologies, the ability of a system to accurately identify anomalous behaviors is incredibly alluring. Continuous evaluation of behavior and health benefit from long-term historical data on daily activity, everyday behaviors, and the quality of movement. The typical setup for such a system involves sensors and devices collecting and producing large chunk of context information in a daily basis. Converting this enormous volume of data into a repository of knowledge using a suitable machine learning approahces so that anomalies can be detected remains a challenging task.

Sensors can be used to detect activity, movement, or health status of an individual. By using different sensors, different types of data can be collected. In the human behavior/activity recognition domain, a wide range of sensors have been utilized and can be generally divided into two groups: obtrusive visual sensors [20] and non-obtrusive sensors [9]. In this project, radar sensors, which are the latter type will be utilized to collect the data for the behavior analysis of an individual due to several advantages over other sensors. Firstly, radar sensors are not privacy intrusive, only location and movement data of an individual will be recorded unlike in vision-based sensors. Secondly, radar sensors do not require a lot of computation and memory. Last but not least, users are not required to wear or carry sensors, the radar sensor will be installed in a fixed location.

Through the utilization of radar sensors, a wealth of data can be provided, including location coordinates, velocity, and acceleration data of an individual. In the healthcare and wellness sector, monitoring of such data obtained in real-time or near real-time enables prompt medical intervention for different scenarios, such as the occurrence of an emergency, treatment and rehabilitation support, or other essential behavioral health assessments related to any form of disease or injury. For example, an increase in frequency of an individual staying at a specific location (sofa/bed) or anomalous patterns in the movement may indicate a possible latent injury or underlying health condition. By analyzing the data obtained from radar sensors, patterns, routines, and deviations from the norm can be unveiled, thereby facilitating informed decision-making.

Radar sensor data is often characterized by its volume, complexity, and noise, necessitating the development of sophisticated algorithms and machine learning techniques to extract meaningful patterns accurately. Therefore, this project will focus on devising efficient algorithms and robust frameworks for collecting, processing, and analyzing radar sensor data to identify individual behaviors and anomalies with valuable behavioral insights.

1.2 Research Aims

This project aims to comprehensively assess the effectiveness of different machine learning/deep learning approaches in capturing human behavior patterns and identifying anomalies within the context of sequential activities and quality of movement. The study will focus on data obtained from radar sensors, which inherently encompass both spatial and temporal features, necessitating a thorough consideration of these factors.

The objectives of this project can thus be summarized as follows:

- Simulate and analyze data needed for recognizing and monitoring human behavior using radar sensors
- Evaluate a range of machine learning/deep learning models implemented for radar sensors with emphasis on incorporating spatial and temporal features
- Analyze the performance of the models in accurately identifying anomalies in the context of sequence of activity and quality of movement.

1.3 Thesis Structure

In this dissertation, chapter 2 will discuss the background theories and insights that underline the methods used in this study. Additionally, related works on anomaly detection for human behavior recognition systems will also be presented. Chapter 3 will discuss the methodology of this project and the process of implementing the models. Chapter 4 will present the results of the project together with detailed discussions on the performance of the models used. Lastly, chapter 5 will conclude the dissertation and provide directions for future works.

Chapter 2

Background and Related Work

Monitoring systems with the proper measurement and communication tools have emerged to support independent living. For instance, sensors can be deployed in houses to track status over time and aid in detecting any abnormalities in an unobtrusive manner. This chapter will provide an overview of existing research and relevant information concerning techniques for representing, recognizing, and detecting anomalies in human behavior.

2.1 Human Behavior Representation

Different methods are utilized to represent and interpret the data extracted from sensors. A sequential pattern identification method, for instance, has been used by the authors of [24] to describe an individual's daily movements. By utilizing this method, activities are illustrated in different columns. Each vertical trajectory (a column) consists of 24 hours and the time spent on each activity are reflected on the clusters in the column.

In [7], a variety of sensors for motion detection, space and storage utilization and appliance use are dispersed around the house to record daily activities. These observations are attributed with start time, duration, weekend/weekday, and activity level. Other than that, [2] introduced the Routine tree, a data structure for describing behavioral patterns. Using data obtained from wearable sensors, the routine tree maps time intervals throughout a day to regular patterns of activity levels at various resolutions.

The authors in [10] proposed the use of IoT-enabled technologies to keep tab on dementia patients. They combined physiological and environmental data to profile and identify changes in participants' health and well-being. Similarly, in [11], to detect urinary tract infection in people with dementia, the authors used environmental data in various locations such as kitchen, bathroom door, chair, bed, and clustered them into four six-hours time interval.

Other than collecting the data from sensors manually, there are works making use of datasets available such as CASAS [12] and ARUBA[1]. These datasets contain sensor data collected from smart homes equipped with various types of sensors, such as motion sensors, door sensors, temperature sensors, and light sensors. The data are then annotated with information about the activities or events that occurred at specific time. The data is typically organized into seqeunces or episodes, where each sequence represents a period of time during which set of activities occurred and can be in the form of minutes, hours, or even days long, depending on the research objectives and the level of granularity required.

2.2 Data Analysis and Anomaly Detection

Since most sensory data is temporally ordered, the classification of the sequences is quite complex. In a variety of fields, sequence classification technique is well-established, including the analysis of genomic data, recognition of gestures and movements, and text data [6]. Extracting features from sequential data is important and challenging at the same time because there are not really any explicit information in the data, and even using a variety of feature selection techniques, it is difficult to turn the sequence into a set of features. Therefore, it is important to analyze and pre-process the sensory data beforehand.

The analysis of sensory data can be divided into two stages [18], the first stage is the lower sensory level, where activities are extracted and classified from the data obtained from sensors. Prior to the classification process, the raw data must first go through different steps, including pre-processing, segmentation and feature extraction. These steps are important. Pre-processing step allows noise, duplication and missing values in the data to be mitigated, segmentation allows the the endless non-uniform temporal ordered data to be divided into equal parts such as using a fixed length sliding window, feature extraction step then allows the data dimension to be reduced by choosing important attributes for the classification process. In the classification process, for data instances with known labels, supervised learning is used and often referred as activity recognition. For unlabelled data instances, unsupervised learning is implemented and this process is known as activity discovery.

Then, higher activity level is the second stage, where the derived activity labels (output from the lower sensory level) are further analyzed from high-level context. Based on the analysis, the act of identifying changes in behavior from an individual's typical living pattern is known as anomaly detection, and it should be noted that the model should be trained accordingly so that abnormal data that has never been seen should be classified or identified as 'abnormal'. Instead of the conventional 2-class (binary) classification problem, this bears the challenge of a 1-class classification problem. In anomaly detection, there are several significant contextual factors in the anomalies such as spatial attributes like location, temporal aspects involving time and duration, sequence of activities, physiological data, movement data. Basically, the task of behavior detection involves modelling an individual's behaviors based on raw sensory data, while anomaly detection pertains to the task of identifying deviations from usual behaviors. Within the realm of monitoring systems, anomaly detection has predominantly found application in ensuring the safety of the elderly and offering assistance to individual with cognitive impairments, such as patients suffering from conditions like dementia [12].

Three distinct methods are employed to determine anomalous data [6]. Firstly, the "Point Anomaly'; approach, which stands as the most straightforward and widely used method, involves setting a predefined threshold value. When an incoming data instance surpasses this threshold, it's classified as an anomaly. Secondly, the "Contextual Anomaly" method makes determinations based on contextual factors like temporal and spatial attributes of the data. Thirdly, the "Collective Anomaly" approach labels a data instance as an anomaly only when considered within a collection, as these instances might not be anomalies if evaluated individually. The outcomes of anomaly detection are conveyed either as a score, represented by the threshold's numerical value, or as a label categorization, classifying instances as "normal" or "anomalous" [8].

2.3 Machine learning/Deep learning approaches

A variety of methodologies have been implemented for the purpose of anomaly detection, encompassing statistical, probabilistic, and machine learning techniques. Over time, diverse strategies have been explored to identify anomalous occupancy patterns within the domain of monitoring systems and smart homes. These approaches span from the extraction of activities from sensory data to the formulation of normal data profiles in instances where samples of anomalous behaviors/activities are infrequent in historical data.

A consensus among many researchers is the assertion that occupancy activities tend to adhere to discernible ”regular patterns,” which could be learned through probabilistic models. The authors in [7] introduced an approach involving individual profiling using a Gaussian Mixture Model to establish the normal data and argued that this method holds an advantage over prior histogram techniques primarily because GMM is adept at capturing interdependencies among attributes. Certain researchers have asserted that the Hidden Markov Model (HMM) finds optimal application within the context of a noisy domain such as smart homes. For instance, authors in [17] utilized HMM as a comprehensive approach for profiling behaviors. By feeding clusters of ”daily activity routines” into a HMM model, they contended that this approach of unsupervised activity extraction was more effective than conventional supervised methods. In contrast, [16] pointed out that the conventional use of HMM as a predictive model to detect anomalous activities has limitations, particularly in identifying similar activity sequences. This is particularly noticeable when the activity duration deviates from the norm, whether being shorter or longer than the typical observed duration.

Authors in [13] highlighted two drawbacks of conventional anomaly detection systems. Firstly, their inability to anticipate future trends, leading to the failure in detecting abrupt disease attacks. Secondly, the reliance on a single context for decision-making contributed to a high rate of false alarms. As a solution, the authors devised an ”integrated system” that harnesses both Hidden Markov Model (HMM) and Fuzzy Logic. This combined approach was designed with the aim of identifying ”multiple contextual activities” and making predictions through aggregating diverse information sources. Within the scope of monitoring daily routines, the authors employed GMM and HMM while incorporating health status as an influential context. This entailed establishing correlations between the health status and daily activities. The resultant system exhibited the capacity to identify anomalies in terms of activity, location, routines (collective anomalies), and variations in health status. Subsequently, the outcomes underwent further analysis through application to the Fuzzy Rule model, culminating in a final prediction.

Lately, there has been a surging interest in the utilization of machine learning and deep learning algorithms within this field. The authors in [19] employed a combination of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) to accomplish multimodal wearable activity recognition. Similarly, [22] delves into the application of CNN for the classification of activities using time-series data collected from smartphone sensors. Building on this, authors in [25] leverages CNN to extract features from raw physiological signals through an unsupervised analysis. Subsequently, anomalies are identified using a multivariate Gaussian distribution, thereby uncovering latent risks. The work outlined in [3] harnessed Recurrent Neural Networks to identify anomalies associated with dementia in everyday living scenarios. While prior efforts in activity recognition based on wearable sensor datasets have highlighted the utility of CNNs and RNNs, there remains ample room for further enhancement in their performance, especially in different scenarios such as different types of sensors used. The same authors also presented [4], where they combined CNN and LSTM and tested on Aruba and CASA.

Chapter 3

Methodology

3.1 Overview

This project aims to apply and evaluate the performance of existing machine learning and deep learning techniques that have been proved to work on different activity/behavioral recognition scenarios on the data collected from a radar sensor. A different model for the same purpose was also suggested aiming to better take into account the nature of the temporal ordered sensory data. Compared to previous work with most of them involving the use of multiple sensors which can detect changes in environmental data in various locations, this project utilizes one radar sensor in a fixed location, the data should thus be represented slightly differently.

This project can be divided into three parts as illustrated in Fig 3.1.

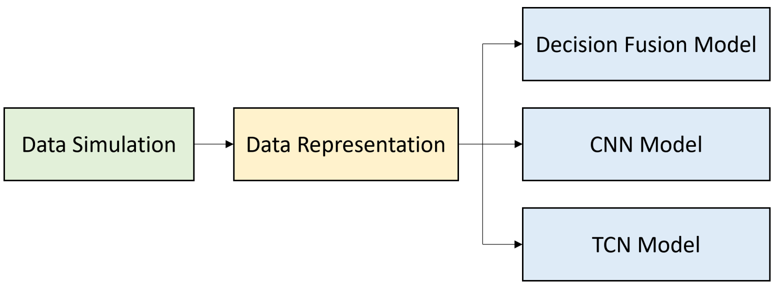


Figure 3.1: Project structure

In the first part of this project, the activity and movement of an individual will be simulated and collected using a radar sensor installed in a fixed location in the room. The collected data are then categorized into normal and abnormal instances. The second part of this project is to pre-process the data into desired format to allow the data to be fed as input to different machine learning/deep learning algorithms. The third part of this project involves detecting anomalous data using different algorithms. Three models are explored in this project: a decision fusion model inspired by [11], a Convolutional Neural Network (CNN) model developed by [4], and a Temporal Convolutional Network (TCN) model. The performance of these models is evaluated based on the accuracy to detect and identify normal and anomalous data.

3.2 Data Representation

The experimental setup for this project is a household room located in Informatics Forum of University of Edinburgh with the dimension of 4 meters x 6 meters as shown in Figure 3.2.

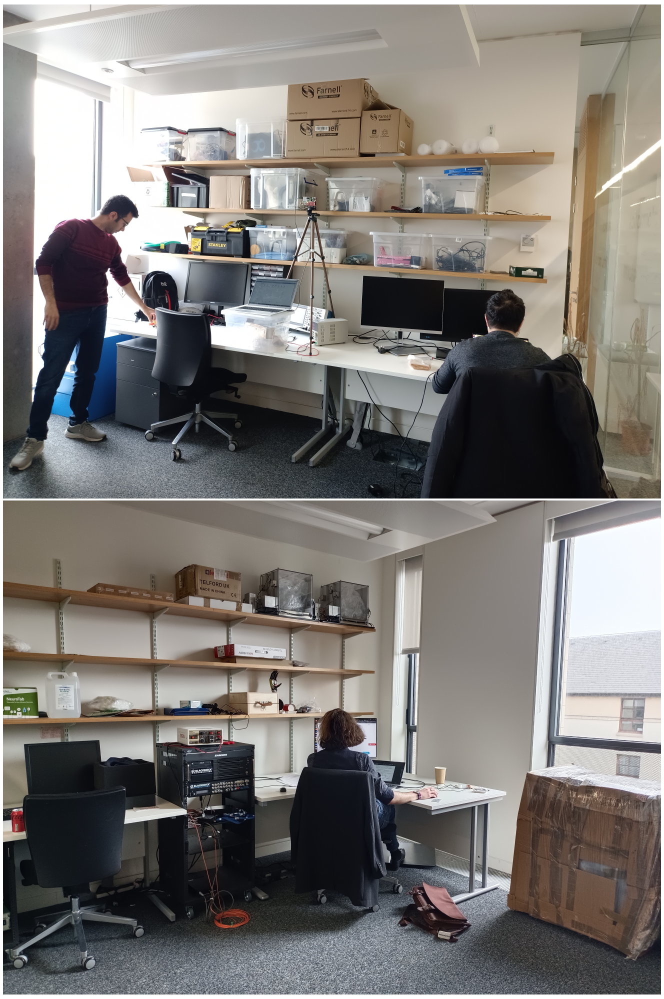
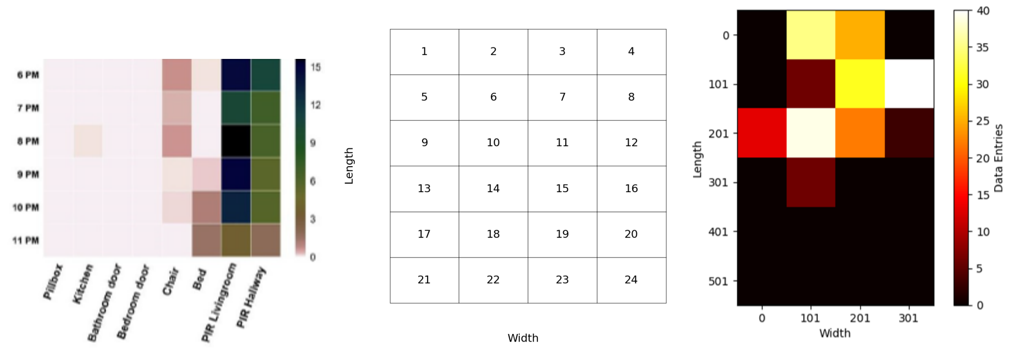


Figure 3.2: Experimental setup for data collection

Activity and movement of an individual in this project will be collected using Texas

Instrument's AWR6843AOP mmWave radar sensor. The sensor will be affixed in a location in the room, which is at the location of the tripod in Figure 3.2. The data entries from the sensor are associated with the movement of an individual, including the x and y coordinates, velocity in x and y axis, acceleration in x and y axis, and timestamps. The sampling rate of the sensor is approximately 12 entries per second, meaning that the timestamp of each entry is recorded with the precision of milliseconds.

Compared to the ways presented in other work for data representation which often involved triggering of sensors in various fixed locations such as kitchen, bed, living room, etc as shown in Figure 3.3a, a more flexible approach is adopted in this project to visualize the activity patterns of an individual according to the dimension of the room instead of predefined locations. For example, given that a dimension of the room is 6m x 4m, the area of the room is $24\,\mathrm{m}^2$. The room is then divided into 24 zones, with each of the 24 zones in Figure 3.3b represents $1\,\mathrm{m}^2$ of the room. The x-axis representing the width of the room and y-axis representing the length of the room. The locations of an individual in a specific time frame can be visualized in a heatmap as shown in Figure 3.3c in which we can see that most time are spent on zone 7. Each heatmap generated represents different time interval. For example, if we have a 5-minute data and the data is segmented into 20 seconds interval, 15 heatmaps like Figure 3.3c are generated. The heatmaps generated are based on the data entries in each time interval, thus approximately 200 entries in each 20 seconds interval.



(a) (b) (c)

Figure 3.3: Different types of data representation (a) Visualization of six-hour sensor firing pattern [11] (b) Zones representing the dimension of the room (c) Example of heatmap generated

3.3 Dataset Construction

In this project, multiple sets of 5-minute data are simulated and captured using a radar sensor. Anomalous behaviors are identified when any behavioral changes occurs including a change in sequence of activities performed by an individual and the movement of an individual.

Instead of identifying specific activities (e.g. sleeping, eating) carried out by an individual, the sequence of activities performed is represented using the zones in 3.3b, for instance, walking around randomly, stopping at an area around zone 13 and 17, stopping at an area around zone 12 and 16, etc.

The normal data and anomalous data are represented in 3.1. In the table, W represents walking in any direction randomly, A represents stopping at a specific location A, and B represents stopping at a specific location B. The numbers in the brackets signify the time spent for each activity in the format of seconds. Thus, the normal data consists of walking around for 60 seconds, stopping at A for 90 seconds, walking around for 30 seconds, stopping at B for 30 seconds, walking around for another 60 seconds, and stopping at B for the last 30 seconds.

A change in sequence of activities and the movement of an individual is the desired anomalies to be identified, thus multiple anomalous data sets are also simulated. Firstly, to identify anomalies in the movement of an individual, while performing the same activity sequence, the individual simulated different walking patterns by imitating the movement of a person with muscle stiffness. Secondly, multiple different activity sequences are also simulated while interchanging the movement pattern. Based on Table 3.1, the first two alphabets of the data set provides indication of the activity sequence while the latter two show the movement pattern. For the activity sequence, NS represents same sequence of activity as the normal data, DS represents distinctly different sequence of activity, and SS represents similar sequence of activity. In SS, the individual performs the same sequence of activity as normal data in the first 3 minutes and different activity sequence for the last 2 minutes. For the movement pattern, NM represents normal movement and walking pattern, and AM represents abnormal movement and walking pattern.

Table 3.1: Data simulated with the sequence of activity and the time spent for each activity in brackets (seconds)

	Sequence of activity
Normal	W(60) → A(90) → W(30) → B(30) → W(60) → B(30)
NSAM	W(60) → A(90) → W(30) → B(30) → W(60) → B(30)
DSNM	W(30) → B(120) → W(30) → A(45) → W(30) → A(45)
DSAM	W(30) → B(120) → W(30) → A(45) → W(30) → A(45)
SSNM	W(60) → A(90) → W(60) → B(60) → W(30)
SSAM	W(60) → A(90) → W(60) → B(60) → W(30)

3.4 Decision Fusion Model

In a real world system, context refers to any detailed information obtained from sensors such as an individual's current or previous activity, location, time spent, and fluctuating physiological data. Thus, the authors in [13] split their model's context space into various context domains, and each context domain is trained using different learning methodologies chosen based on the properties of the domain's context information. The results from each domain are then combined and connected using fuzzy model to ultimately create context-aware decisions.

The first step in detecting anomalies is to construct a model based on normal observations. New observations are then compared to the normal data and deviation is calculated. Similar to [13], the first context domain of the decision fusion model in this project is the individual's location/activity and the purpose is to detect anomalous behaviors in the sequence of activities carried out. The second context domain of the model is the individual's movement pattern. Deviations in movement patterns may presage health problems and as a result, early detection of such issues can mitigate health risks. Fig depicts the entire system architecture for the decision fusion model.

3.4.1 Sequence of activity

For the first context domain, using the Markov chain model developed by [10], to analyze the daily activities and detect anomalies based on sequence of activities, the authors implemented a Markov chain model to represent the activity sequences and an 'entropy rate' is computed to assess the regularity of the sequence of activity.

3.4.1.1 Markov chain

In a normal daily life, each activity carried out in subsequent manner is related, it is thus important to profile the sequence of activity using a probability-based model in which the probability of each activity is dependent on the previous step but not the entire sequence [23]. To do so, the sequence of activities are modeled as a Markov chain,

$$P(x_i|x_{i-1},...,x_1) = P(x_i|x_{i-1}) = p_{i-1,i} \tag{3.1}$$

in which $p_{i-1,i}$ is the transition probability of $x_{i-1} \rightarrow x_i$ with x representing different activity. Each chain is thus represented as a Markov model with a corresponding transition matrix with Figure 3.4 showing an example of a Markov model with states a, b, c, and d.

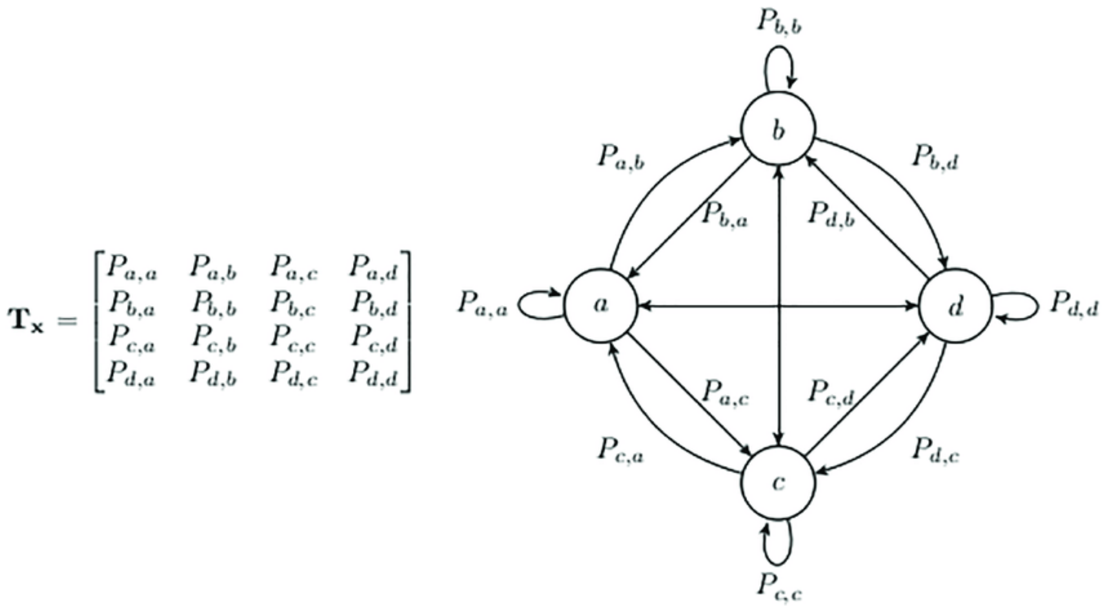


Figure 3.4: A sample Markov model with 4 states [10]

3.4.1.2 Entropy rate

Entropy has been widely researched for pattern recognition tasks and is a popular measure for quantifying information and reflecting the degree of randomness in a sequence [21]. The entropy rate value incorporating the transition probabilities between different states $(P_{\alpha\beta})$ and the state probabilities (P_α) is defined as [14]:

$$\zeta = -\sum_{\alpha\beta} P_\alpha P_{\alpha\beta}.log(P_{\alpha\beta}) \tag{3.2}$$

3.4.1.3 Clustering techniques

Before modeling a Markov chain, the data in each time frame in the form of Figure 3.3c should first be identified and clustered into three different groups. The three clusters should represent stopping at location A, stopping at location B, and walking around randomly. To classify the information from the radar sensor, different clustering techniques are implemented and compared.

Two types of clustering techniques are used. Firstly, K-means clustering segments n observations into k clusters by deducing the nearest mean to each cluster [26]. Secondly, hierarchical clutering tehcnique creates a hierarchical structure of clusters by iteratively merging or splitting data points based on their similarity.

3.4.1.4 Algorithm and anomaly detection

Multiple normal 5-minute data is first split into training sets and verification sets. If each data is divided into different subsets of 20-second intervals, each 5-minute data is represented by 15 slices of data. The x and y coordinates are then extracted and a heatmap for each slice are generated. Based on the heatmaps, clustering techniques are implemented to map each slice of data to a single state. These states are then used to model a Markov chain. The transition probabilities are calculated and the entropy rate value is deduced. After obtaining the entropy rate value for the training set, an entropy rate value for each verification set is also calculated and used to determine the deviation bounds and thus forming the confidence interval δ:

$$\delta = \zeta_T \pm \mu \frac{\sigma}{\sqrt{v}} \quad where \quad \sigma = \sqrt{\frac{\sum(\zeta_V - \zeta_T)^2}{v}} \tag{3.3}$$

Using the equation and confidence interval computed, anomalous behaviors are detected with entropy rate values out of the range of the confidence interval. It should be noted that the centroids of the clusters obtained from the clustering techniques are constant even for the verification sets and test sets as the clusters are only predicted using the training set and the same model is then utilized, meaning that each activity is represented with the same cluster throughout the whole process.

3.4.2 Movement pattern

For the second context domain, the movement data from the sensor are taken into account here, which are velocity and acceleration. Based on the clusters identified, only the data from W clusters (walking around randomly) are extracted. Velocity and acceleration magnitude for each data entries are calculated by taking the square root of the sum of squaring both the x and y components of velocity and acceleration, respectively. These data points are then used to model a Gaussian distribution. The Gaussian Mixture Model (GMM) algorithm then estimates the parameters (mean and covariance) of the velocity and acceleration Gaussian distribution.

Once the GMM is trained, when new data are introduced, the likelihood of observing each data point under the GMM model is calculated for each Gaussian component. Log-likelihood is used and it quantifies how well the data point fits each Gaussian distribution. If the log-likelihood of a data point is significantly lower than expected (based on the modeled distribution of normal data), it suggests that the data point is less likely to be part of the modeled distribution. Data points with low log-likelihoods are considered as anomalies, as they deviate from the normal behavior captured by the GMM.

3.4.3 Anomaly detection

To detect abnormal instances, the output from both model will be represented as “0” or “1”, indicating “normal” and “abnoraml” respectively and sent to a decision layer as shown in Figure 3.5.

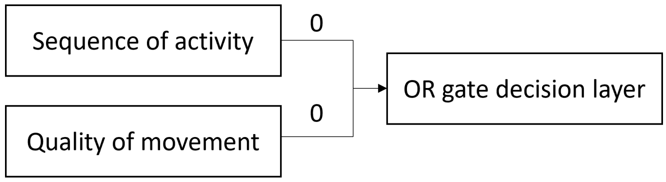


Figure 3.5: Decision fusion model

The decision layer will act as an OR gate making decisions as shown in Table 3.2.

Table 3.2: OR gate decisions

Activity sequences	Quality of movement	Decision
0	0	0
0	1	1
1	0	1
1	1	1

3.5 Convolutional Neural Network (CNN)

Instead of using separate models to identify anomalies in different context domains and fusing them, the effectiveness of a single model to identify anomalous behaviors based on both sequence of activities and movement patterns is also investigated. In other words, using normal data as training set, the model should be able to identify NSAM, DSNM, DSAM, SSNM, SSAM as anomalies.

The idea of using CNN is explored with CNN models being particularly effective at learning spatial features and patterns from two-dimensional data. The CNN model is trained to recognize the normal behaviors and encode the normal behavior routines with the movement pattern. The trained model is then utilized to detect anomalies deviating from the normal behaviors, including both sequence of activity and movement patterns.

3.5.1 Data Preparation

The CNN model developed in [4] for daily life recognition and abnormal behavior detection tasks is slightly modified and implemented. To feed the data into a CNN model, it should be pre-processed into desired format. The normal data are first split into training set and verification set. For each of the 5-minute training set, 15 heatmaps (20 seconds each) are generated based on the zones as shown previously in Figure 3.3b and Figure 3.3c. Each 6 x 4 data (heatmaps) are then flattened into a single 1 x 24 data, resulting in 15 1 x 24 flattened data as shown in Figure 3.6.

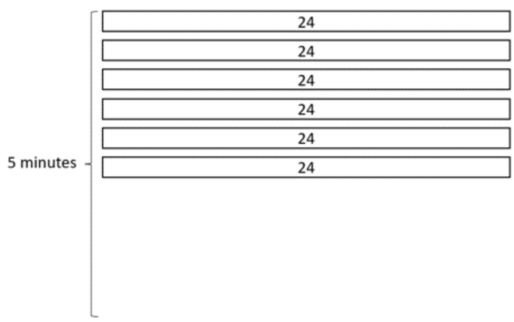


Figure 3.6: Flattened data (sequence of activity)

Figure 3.7 visualizes how the data look like with each column in the 15 heatmaps representing one of the 24 zones.

The next step is to aggregate the movement pattern data each array. For each heatmaps, movement data are concatenated as features as illustrated in Figure 3.8. From the radar sensor, information related to the movement data that are extractable are the

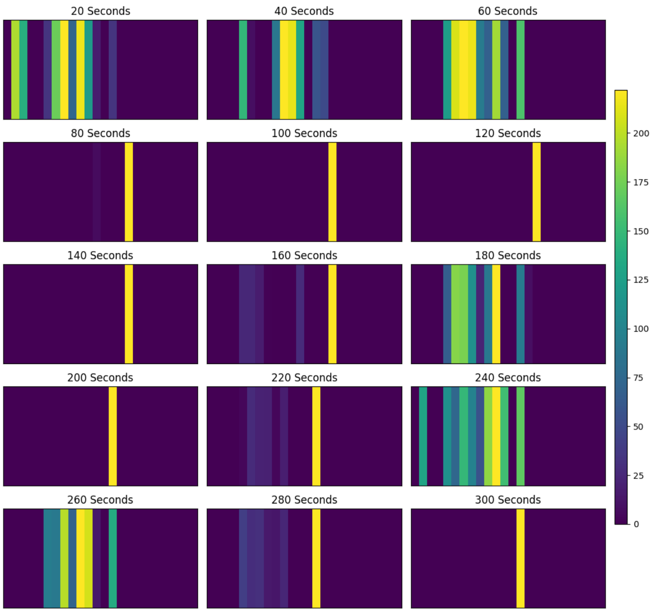


Figure 3.7: Representation of the flattened data for every 20 seconds

velocity and acceleration in both x and y axis. Velocity and acceleration magnitude can be calculated by using the equations:

$$v_{mag} = \sqrt{(v_x)^2 + (v_y)^2} \tag{3.4}$$

$$a_{mag} = \sqrt{(a_x)^2 + (a_y)^2} \tag{3.5}$$

The velocity and acceleration direction can be calculated using the equations:

$$\theta_v = \arctan\left(\frac{v_y}{v_x}\right) \tag{3.6}$$

$$\theta_a = \arctan\left(\frac{a_y}{a_x}\right) \tag{3.7}$$

In this case, movement features for every 20 seconds are calculated and concatenated here, including velocity magnitude mean, acceleration magnitude mean, velocity magnitude standard deviation, acceleration magnitude standard deviation, velocity magnitude median, acceleration magnitude median, dominant velocity direction, and dominant acceleration direction.

After concatenating the movement data, each 5-minute data can be represented in a single 2-dimensional array by vertically stacking every 20-second time interval heatmaps as shown in Figure 3.9, with the x-axis representing 32 features (24 zones + 8 movement features), and y-axis representing different 20-second time window (15 in total). In other words, Figure 3.9 is formed by stacking all the flattened data from

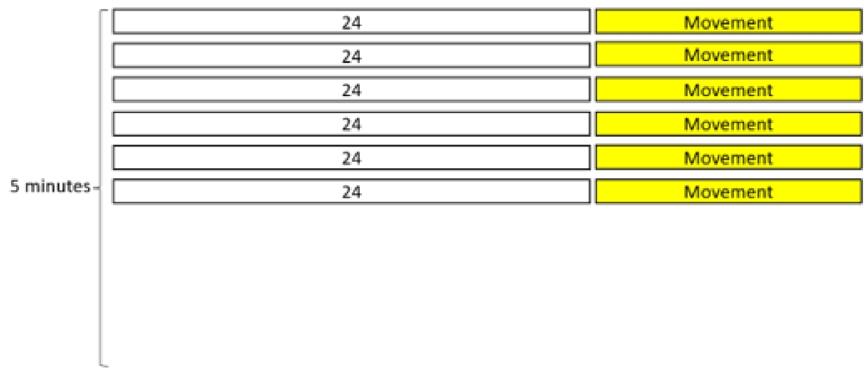


Figure 3.8: Flattened data (sequence of activity + quality of movement)

Figure 3.7 vertically. It should also be noted that the 24 spatial features range from 0 to 250, and the 8 movement features range from -4 to 4, the movement features are thus normalized to the range of -250 to 250 to avoid exploding gradients.

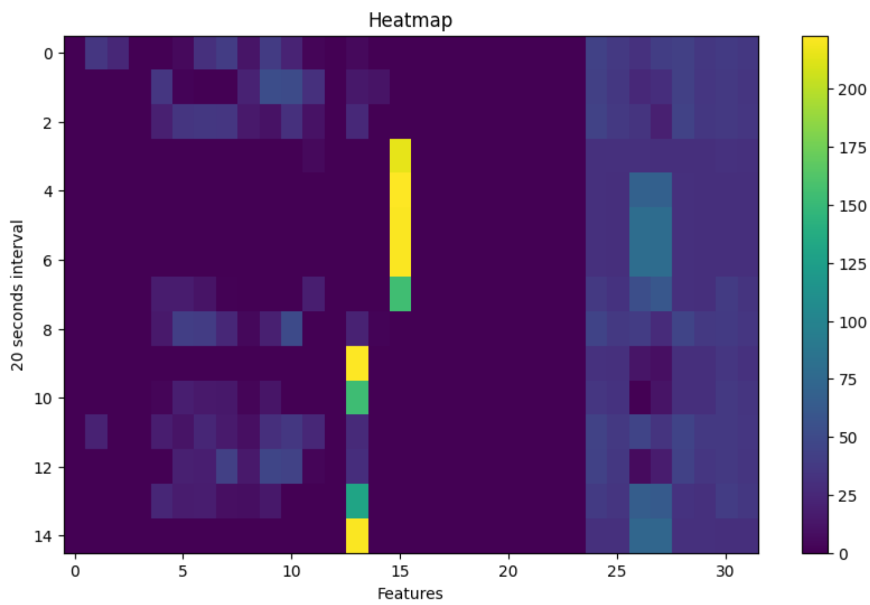


Figure 3.9: Data representation of a single 5-minute data

3.5.2 Anomaly detection

Since the input data consists of multiple (15, 32) matrices, the first step is to normalize the data to be in the range of [-1,1]. The CNN model accepts inputs with the following dimensions: h x w x d, the input shape of the data is thus reshaped to be (15, 32, 1), with 15 and 32 being the height x weight corresponding to the original shape, and 1 is the channel because the data is grayscale (single channel). The CNN model is designed with two 2D convolution layers to learn the spatial patterns and features. The first 2D convolution layer has 32 filters with a kernel size of (6,6), followed by an activation function. The second layer has 64 filters with the same kernel size and activation function. A max-pooling layer is followed by each convolution layer to reduce the spatial dimensions to a single value per channel. The output is then fed to the dense layer with 480 units (15 x 32), which will then be fit to a reshape layer, which flattens the output back into a 2D grid with the same dimensions as the original input. The model is trained using the normalized training data. The training process involves optimizing the model's parameters (weights and biases) using the Adam optimizer and minimizing the mean squared error loss. The architecture is illustrated in Figure 3.10.

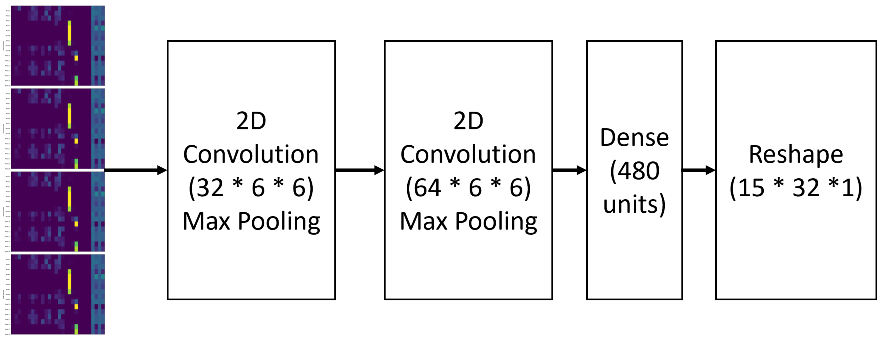


Figure 3.10: Architecture of the CNN model

To detect anomalies, the trained model is used to reconstruct the verification set and test set. Both set are normalized using the same normalization factors as the training data and the model predicts the reconstructed output for both. The reconstruction error for each data is calculated as the mean squared error between the normalized verification set/test set and the reconstructed output. From the verification set, the highest error is selected to be the threshold. If a data point from the test set has a reconstruction error greater than the threshold, an anomaly is detected.

3.6 Temporal Convolutional Network (TCN)

While CNN models perform well in learning spatial patterns, they are not ideal for modelling sequential data with temporal features. Therefore, CNN models are often paired with Recurrent Neural Network (RNN) such as Long Short Term Memory [27] for such purpose.

In this section, a Temporal Convolutional Network (TCN) model is implemented to investigate the performance of such model concerning the nature of the data containing spatial and temporal features.

3.6.1 Convolutional sequence model

TCN was introduced in [5]. TCN's temporality and adaptable receptive fields make it ideal for modelling sequential data due to casual convolutions [15]. The TCN has two fundamental limitations: it may only make use of information from previous timesteps and its output should have the identical shape as its input. 1D fully-convolutional network design is employed in TCN to satisfy these temporality criteria so that all of its convolution layers have the same length and zero padding ensures that higher levels (present) have the identical length as lower layers (past).

Additionally, TCN employs causal convolutions, wherein each layer's output is computed at time step t using only the region computed at time step t or in layers from previous timesteps, as shown in Figure 3.11.

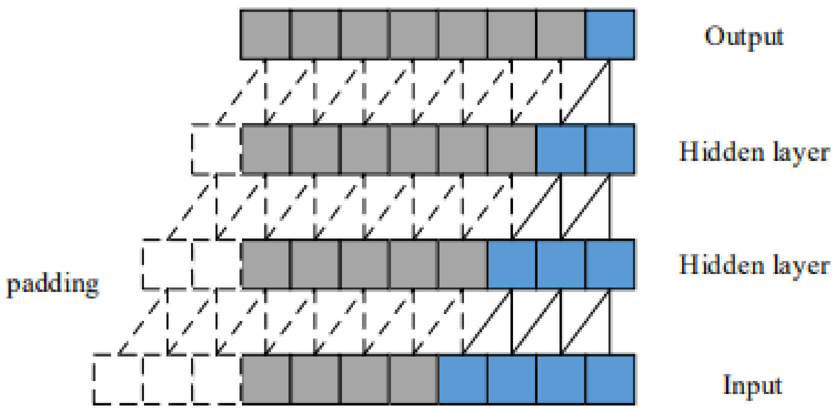


Figure 3.11: A casual convolution with filter kernel size k=2 [15]

3.6.2 Anomaly detection

The data is pre-processed similarly to the data fed into the CNN model with the size of (15,32) for each data, indicating 15 timesteps and 32 features. The TCN model starts with three 1D convolutional layers, which perform convolutions over the temporal dimension (15 timesteps). Each convolutional layer has a kernel size of 6 and the 'causal' padding, which ensures the filters do not peek into future timesteps during training, which is crucial for processing sequential data.

After each convolutional layer, a *tanh* (hyperbolic tangent) activation function is applied element-wise to the output of the convolution. After the convolutional layers, a dense layer with 32 neurons aggregates the information learned from the convolutional layers and produces a representation with 32 features for each timestep. The output of the TCN model is the reconstructed sequence with the shape of (15,32). Once the TCN is trained on the training set, steps similar to the CNN model are applied for anomaly detection.

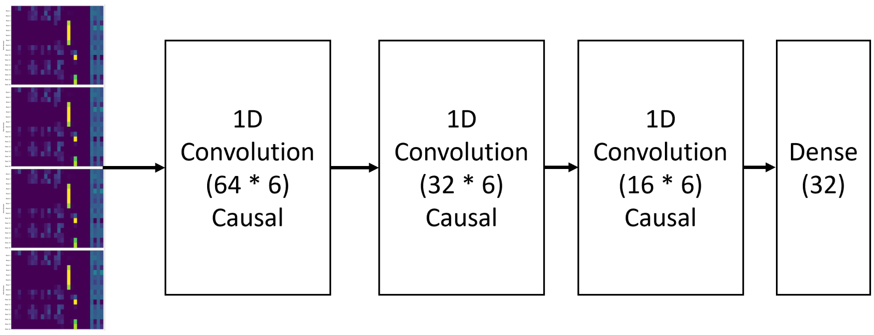


Figure 3.12: Architecture of the TCN model

3.6.3 Evaluation Method

In this project, the outcomes predicted by the model are divided into four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The terms ”true” and ”false” indicate whether the model’s predictions are accurate, while ”positive” and ”negative” refer to the predicted categories. ”Positive” signifies the detection of anomalies, while ”negative” indicates normal data.

These values are used to evaluate the models’ overall effectiveness through various calculations:

- Accuracy: Calculated as the ratio of correctly classified outcomes (TP + TN) to the total number of predicted outcomes (TP + TN + FP + FN). A higher accuracy score suggests better classification performance.

- Recall: Calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN), which is the total number of actual positive instances. A high recall indicates that the model is effective at capturing most of the positive cases in the dataset, minimizing the chances of missing out on true positives.

- Precision: ssesses the accuracy of the model’s positive predictions. It calculates the ratio of true positives (TP) to the sum of true positives and false positives (FP), which is the total number of instances predicted as positive by the model.

Precision gives an indication of how well the model avoids false positives – instances that were predicted as positive but are actually negative.

Chapter 4

Results and Discussion

4.1 Data Representation

13 normal data are simulated and generated, 21 abnormal data are simulated and generated according to the activity sequence and the movement pattern specified in Table 3.1. The abnormal movement and walking pattern is simulated by a normal individual imitating the movement pattern of an individual with muscle stiffness.

Based on Figure 4.1, the sensor is located around the top of the edge of zones 2 and 3.

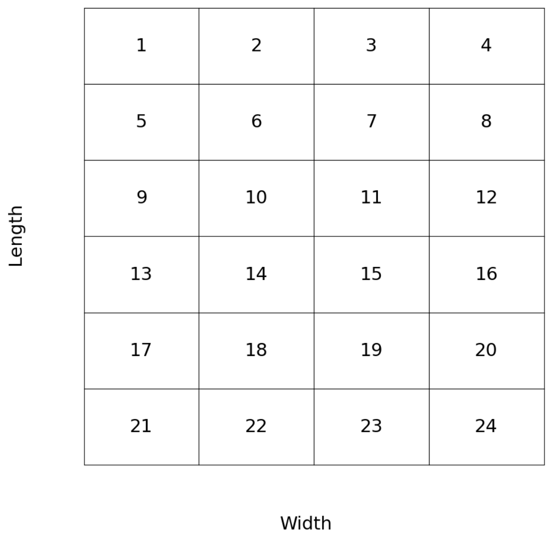
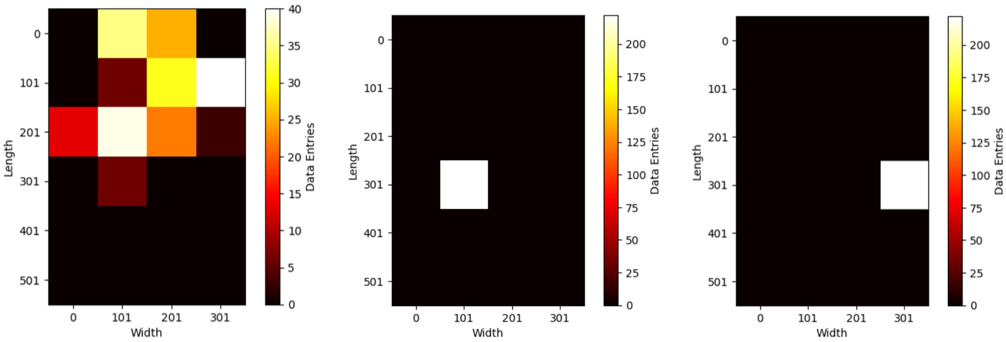


Figure 4.1: Zones of the room

Walking around randomly involves the individual moving around randomly in the room without any restriction, location A is around zone 13 and 14, and location B is around zone 15, 16, 19, and 20. The heatmap generated for each situation can be visualized in Figure 4.2.



(a) (b) (c)

Figure 4.2: Heatmap representing (a) Walking around randomly (b) Stopping at location A (c) Stopping at location B

To generate the heatmaps in Figure 4.2, we must first decide the time interval to slice the data. Based on the simulated data, the minimum time spent in any activity is 30 seconds, thus it should be less than 30 seconds. As we are working on multiple 5-minute data in this research, we should take into account the enormously large data produced if such system is deployed in real time which will be represented in the form of minutes, hours, or even days. The time interval should be small enough for us to identify different activities with clear boundaries while not too small that will result in significant amount of data. Therefore, we picked 20 seconds to be the ideal time interval in this project.

4.2 Decision fusion model

4.2.1 Clustering techniques

To evaluate the accuracy of clustering techniques used, the expected clusters and the identified clusters from 5 sets of normal data are compared and presented in Table 4.1.

Based on the data sliced into 20 seconds for each time interval, the expected clusters are as follows with 0 as the walking around randomly cluster, 1 as the stopping at location A cluster, and 2 as the stopping at location B cluster:

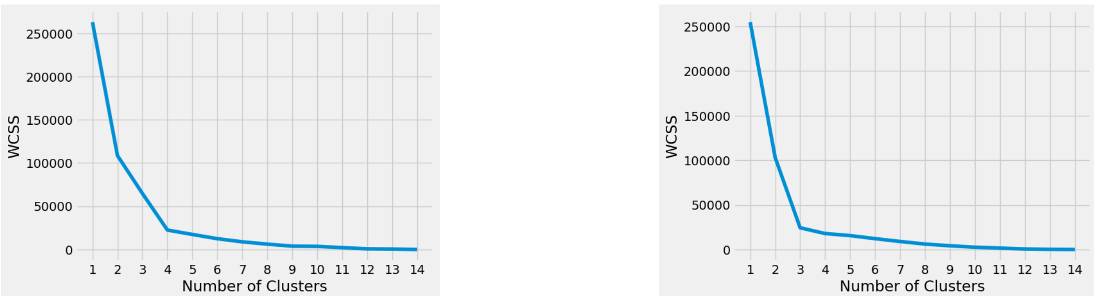
Expected: 0 0 0 1 1 1 1 0/1 0 2 0/2 0 0 0/2 2

As shown in Table 4.1, the hierarchical clustering technique outperforms K-means clustering in this case, with 100% accuracy vs 96% accuracy. To understand why, we

Table 4.1: Accuracy comparison between K-means and Hierarchical Clustering

	Clustering techniques	Accuracy
Normal Data 1	KM: 0 0 0 1 1 1 1 1 0 **0** 2 0 0 0 2	14/15
	HC: 0 0 0 1 1 1 1 1 0 2 2 0 0 0 2	15/15
Normal Data 2	KM: 0 0 0 1 1 1 1 1 0 2 2 0 0 0 2	15/15
	HC: 0 0 0 1 1 1 1 1 0 2 2 0 0 0 2	15/15
Normal Data 3	KM: 0 0 0 1 1 1 1 1 0 **0** 2 0 0 0 2	14/15
	HC: 0 0 0 1 1 1 1 1 0 2 2 0 0 0 2	15/15
Normal Data 4	KM: 0 0 0 1 1 1 1 0 0 2 2 0 0 0 2	15/15
	HC: 0 0 0 1 1 1 1 1 0 2 2 0 0 0 2	15/15
Normal Data 5	KM: 0 0 0 1 1 1 1 1 0 2 2 0 0 0 **0**	14/15
	HC: 0 0 0 1 1 1 1 1 0 2 2 0 0 0 2	15/15

used two of the normal data and plotted the elbow plot for K-means clustering in Figure 4.3, which is a graphical method used to determine the optimal number of clusters in a K-means clustering algorithm. The K-means algorithm aims to minimize the within-cluster sum of squares (WCSS), which is a measure of the variance within each cluster. As you increase the number of clusters, the WCSS tends to decrease because each point gets closer to its cluster's centroid. The optimal number of clusters is often chosen at the point where the decrease in within-cluster sum of squares (WCSS) slows down, forming an "elbow" in the plot. This signifies a point of diminishing returns, where adding more clusters doesn't provide a significant gain in variance explained.



(a) (b)

Figure 4.3: Elbow plots of (a) Normal data 1 (b)normal data 2

We can see that the optimal number of clusters fluctuates between 3 and 4 based on the elbow plots. K-means algorithm aims to partition the data into the predetermined number of clusters (in this case, 3), assigning each data point to the cluster whose centroid it is closest to. However, in scenarios where the centroids of neighbouring clusters are relatively close, misclassification can occur. For instance, if cluster 0 has a centroid around zone 15, cluster 1 around zone 17, and cluster 2 around zone 20, a new data point near zone 16 (which is expected to belong to cluster 2) might be erroneously assigned to cluster 0 due to its closer distance to the centroid of cluster 0. This highlights the sensitivity of K-means to initial centroid placements and the distribution of data points within the feature space, which can lead to misclassification in cases of clusters with overlapping centroids. On the other hand, hierarchical clustering technique does not require the number of clusters to be specified, it creates an hierarchy of clusters by iteratively merging or splitting them based on similarity. Consequently, when clusters have overlapping or irregular shapes, the hierarchical approach can provide a more accurate representation of the data's underlying structure compared to the rigid partitioning of K-means.

4.2.2 Results

In this analysis, for the first context domain (sequence of activities), we initially divided the 13 sets of normal data into 5 training sets, 3 verification sets, and 5 test sets. Each 20-second window in a set of data is mapped to different clusters using hierarchical clustering technique, essentially creating representative states. With the cluster-based states established, we proceeded to calculate the transition probabilities within each set. This involved not only the normal data, but also encompassed all the test data, comprising 5 sets each of NSAM, DSNM, and DSAM, along with 3 sets each of SSNM and SSAM.

The central metric we evaluated was the entropy rate, a measure of uncertainty, which was computed for each set. By deriving confidence intervals for the entropy rate through analysis of the training and verification sets, we then gauged the statistical significance of the entropy values calculated for the various test sets.

For test sets where the entropy rate fell within the established confidence interval, we interpreted this as a ”normal” behavior and dispatched a corresponding signal (“0”) to the OR gate decision layer. Conversely, if the entropy rate exceeded the confidence interval range, we deemed the condition as ”abnormal” and transmitted a signal (“1”) to the decision layer.

For the second context domain, which revolves around evaluating the quality of movement, our focus was exclusively directed towards the clusters associated with

Table 4.2: Anomaly detection results

	Sequence of activity	Quality of Movement	Decision
Normal Data 1	0	0	0
Normal Data 2	0	0	0
Normal Data 3	1	1	1
Normal Data 4	0	0	0
Normal Data 5	1	0	1
NSAM 1	0	1	1
NSAM 2	1	1	1
NSAM 3	0	1	1
NSAM 4	1	1	1
NSAM 5	1	1	1
DSNM 1	1	0	1
DSNM 2	1	0	1
DSNM 3	1	1	1
DSNM 4	1	0	1
DSNM 5	1	0	1
DSAM 1	1	1	1
DSAM 2	1	1	1
DSAM 3	1	1	1
DSAM 4	0	1	1
DSAM 5	1	1	1
SSNM 1	1	0	1
SSNM 2	0	0	0
SSNM 3	1	0	1
SSAM 1	1	1	1
SSAM 2	0	1	1
SSAM 3	0	1	1

”walking around randomly.”. Within this scope, our analysis took into account movement data specifically recorded during walking activities.

To characterize these movements, we extracted the data embedded in the velocity and acceleration components along the x and y axes. By processing these parameters, we derived the magnitudes of both velocity and acceleration for individual data entries present within the ”walking around randomly” cluster.

Extracting these data from the 8 training sets (5 training sets and 3 verification sets), a GMM is trained and used to identify whether the 23 test sets are normal or abnormal by calculating the log likelihood. A threshold is established by averaging the log-likelihood values from the two verification sets.

In the final step of this analysis, we categorized test sets based on their log-likelihood values. Sets with log-likelihood values falling below the threshold were classified as “abnormal”, signifying a departure from expected movement quality. In such instances, a signal (“1”) was conveyed to the decision layer. Conversely, test sets with log-likelihood values exceeding the threshold were regarded as “normal”, marked by the absence of an abnormality signal.

The decisions in each layer of the decision fusion model are presented in Table 4.2.

The performance of the decision fusion model can be summarized as shown in Table 4.3. Notably, the model exhibits a high accuracy of 95.65%, primarily attributed to the GMM’s adeptness in effectively flagging anomalies within the quality of movement context, which only misclassifed one normal movement as “abnormal”, thus resulting in high accuracy for the OR gate decision layer.

Table 4.3: Performance of decision fusion model

	Accuracy	Precision	Recall
Decision Fusion Model	0.9565	0.9091	0.9524

By employing a Markov chain model to model the sequence of activity, we can see that the margins for error are very tiny, if there is only a slight disparity or near identical in the sequence of states (activities identified) between the training and verification sets and if the test set have a slightly more different sequence of states, the entropy value for the test set will be outside the confidence interval even if it is a normal sequence of activity.

Other than that, the Markov chain model does not have the ability to distinguish more complex forms of anomalies. For example, if we have two same sequence with locations swapped, the model’s reliance on transition probabilities results in an identical entropy rate calculation, thereby failing to flag such instances as “abnormal”. To further explain this, given the scenario of two same sequence with locations “A” and “B” swapped, the state probability (P_A) and the transition probability related to location “A” $(e.g., P_{AW})$ in the first sequence (normal data) will be the same as the state probability,

(P_B) and the transition probability related to location "B" ($e.g., P_{BW}$) in the second sequence (anomalous data). In a real life scenario, we would want to identify this scenario as "abnormal" as it could signify that a person is spending more time at a resting place more than working space.

It should also be noted that only velocity magnitude and acceleration magnitude are relevant metrics for the GMM. Notably, the introduction of additional metrics to better quantify movement quality might result in a reduction in accuracy.

4.3 CNN model

During the experiments for the CNN model, the optimization technique employed was Adam, known for its ability to facilitate improved convergence of the model. The Mean Squared Error Loss functon was utilized to quantify the disparity or error between the reconstructed sequence and the test sequence. We investigated the impacts of varying the number of epochs, the kernel size, and the activation function.

Throughout the training process, as the number of epoch increases, the model's performance transitions from underfitting (poor performance on the training set)to overfitting (good performance on the training set but struggles to generalize to new data). We used different numbers of epochs, including 50, 100, and 200. The kernel size directly impacts how the CNN model captures features (higher-level or more fine-grained) from the training data. The kernel used is (3,3), (6,6), and (8,8). Activation functions introduce non-linearity to the network, enabling it to learn complex relationships in the data. Different activation functions are used, including Rectified Linear Activation (ReLU), Sigmoid, and Tanh (hyberbolic tangent).

4.3.1 Number of epochs

The results of using different number of epochs are shown in Table 4.4 . The kernel size is set to be (6,6) and the activation function is Tanh. By spltting the training data into training set and validation set, the training loss and validation loss will be the measures used to validate the results.

4.3.2 Kernel size

The results of using different kernel size are shown in Table 4.5 . The number of epochs is set to be 100 and the activation function is Tanh. By varying the kernel size, we can

Table 4.4: Number of epochs experiment (CNN)

	Epochs	Training Loss	Val Loss
Normal data	50	0.0045	0.0092
Normal data	100	0.0028	0.0088
Normal data	200	0.0027	0.0075

vary the quantity of features to take into account at a time together with the number of timesteps.

Table 4.5: Kernel size experiment (CNN)

	Kernel size	Training Loss	Val Loss
Normal data	(3,3)	0.0043	0.0098
Normal data	(6,6)	0.0015	0.0071
Normal data	(8,8)	0.0019	0.0075

4.3.3 Activation function

The results of using different activation functions are shown in Table 4.6 . The number of epochs is set to be 100 and the kernel size is set to be (6,6).

Table 4.6: Activation function experiment (CNN)

	Kernel size	Training Loss	Val Loss
Normal data	ReLU	0.0042	0.0096
Normal data	Sigmoid	0.0038	0.0071
Normal data	Tanh	0.0029	0.0070

4.3.4 Performance

Based on the results of using different hyperparameters, the best hyperparameters for the CNN model are 200 epochs, (6,6) kernel size, and Tanh activation function. Using the highest reconstruction error for the verification set, the threshold are set to be 0.0050. The reconstruction error for each test data are deduced and the range of the reconstruction error for each set are shown in Table 4.7.

Table 4.7: Anomaly detection results (CNN)

	Range of reconstruction error	Accuracy
Normal Data	0.0040 - 0.0052	4/5
NSAM	0.0048 - 0.0078	3/5
DSNM	0.043 - 0.069	5/5
DSAM	0.080 - 0.095	5/5
SSNM	0.0013 - 0.018	2/3
SSAM	0.017 - 0.020	3/3

Table 4.8: Performance of CNN model

	Accuracy	Precision	Recall
Decision Fusion Model	0.8518	0.9500	0.9048

4.4 TCN model

Similar to the CNN model, the TCN model was trained using Adam optimization parameter and mean squared loss function was utilized. Same sets of parameters are evaluated. With TCN model employing 1D convolutional layers, the kernel size used are adjusted to be 3,6, and 8 instead.

4.4.1 Number of epochs

The results of using different number of epochs are shown in Table 4.9 . The kernel size is set to be 6 and the activation function is Tanh.

Table 4.9: Number of epochs experiment (TCN)

	Epochs	Training Loss	Val Loss
Normal data	50	0.0035	0.0089
Normal data	100	0.0015	0.0080
Normal data	200	0.0009	0.0060

4.4.2 Kernel size

The results of using different kernel size are shown in Table 4.10 . The number of epochs is set to be 100 and the activation function is Tanh.

Table 4.10: Kernel size experiment (TCN)

	Kernel size	Training Loss	Val Loss
Normal data	3	0.0011	0.0041
Normal data	6	0.0007	0.0028
Normal data	8	0.0008	0.0029

4.4.3 Activation function

The results of using different activation functions are shown in Table 4.11 . The number of epochs is set to be 100 and the kernel size is set to be 6.

Table 4.11: Activation function experiment (TCN)

	Kernel size	Training Loss	Val Loss
Normal data	ReLU	0.0022	0.098
Normal data	Sigmoid	0.043	0.041
Normal data	Tanh	0.0007	0.0028

4.4.4 Performance

Based on the results of using different hyperparameters, the best hyperparameters for the TCN model are 200 epochs, kernel size of 6, and Tanh activation function. Using the highest reconstruction error for the verification set, the threshold are set to be 0.0045. The reconstruction error for each test data are deduced and the range of the reconstruction error for each set are shown in Table 4.12.

Table 4.12: Anomaly detection results (TCN)

	Range of reconstruction error	Accuracy
Normal Data	0.0019 - 0.0038	5/5
NSAM	0.0065 - 0.0099	5/5
DSNM	0.030 - 0.054	5/5
DSAM	0.067 - 0.083	5/5
SSNM	0.009 - 0.010	3/3
SSAM	0.009 - 0.017	3/3

Table 4.13: Performance of CNN model

	Accuracy	Precision	Recall
Decision Fusion Model	1.0	1.0	1.0

4.5 CNN vs TCN model

By comparing the performance of the CNN and TCN model, it is evident that the TCN model outperforms the CNN model in detecting anomalies using the data collected.

In the case of the CNN model, through convolution, the spatial features are taken into account based on the arrangement of the features. However, this can lead to misclassifications due to the representation of data. For instance, zones like 7, 11, and 15 are adjacent in the physical layout but separated when the data is flattened, might be misinterpreted. Furthermore, This limitation results in the model misclassifying normal data instances and struggling to identify anomalies when there are slight deviations in sequence or distinct movement patterns.

Conversely, the TCN model takes both temporal and spatial features into account effectively. It processes multiple sequence of 32 data points (spatial + movement attributes) as input. By moving in one dimension, temporal features are taken into account by sliding the filters over the sequence. Notably, the TCN model demonstrates the potential to quantify the degree of data abnormality, as each type of anomalous data can be discerned through specific ranges of reconstruction errors.

In summation, the TCN model's ability to incorporate both temporal and spatial features, coupled with its potential to quantify abnormality, underscores its superior performance over the CNN model in anomaly detection based on the collected data.

Chapter 5

Conclusions

5.1 Summary

In conclusion, this project has effectively demonstrated the viability of using radar sensor in the domain of human activity recognition/monitoring where previous studies have mainly focused on using different types of sensors in this domain. In this project, human behavior is modeled in the context of activity sequence based on time spent on different locations and the quality of movement. By using a radar sensor to collect data for human behavior recognition, the data are represented differently compared to previous approaches by segmenting the dimension of the room into different zones and generating heatmaps based on the movement/activity.

Two different existing machine learning methods (Decision fusion model and CNN model) were applied to the data collected in a household room. To better take into account the temporal features of the radar sensor, a TCN model was also constructed and tested on the data.

For the decision fusion model using an OR gate decision layer, attributed to the adeptness of the GMM model to identify anomalous data for movement patterns, the final decision registers a 95.65% accuracy. However, the Markov chain model does not accurately detect anomalies in the context of sequence of activity due to the fact that it relies on the entropy rate value deduced from the transition probabilities.

The performance of using a single model to detect anomalies in both context domain is also investigated. The CNN model achieved a 85.18% accuracy. The CNN model struggles to detect anomalies on data with same sequence of activity and anomalous movement pattern, or marginally dissimilar sequence of activity due to the fact that the CNN model does not take into account the temporal information well. To better take into account the temporal feature, a TCN model is used. It achieved a 100% accuracy based on the data generated. By tweaking the size of the filter, the TCN model can determine how many adjacent time steps to take into account. In other words, a larger filter size allows the TCN to capture longer-range dependencies and patterns in the input time series data. A smaller filter size captures shorter-range patterns and works well for capturing local fluctuations in the data.

5.2 Future work

There is still much room for improvement in this project. Firstly, the data imbalance issue should be addressed and more normal data should be collected to verify the performance of the models. Secondly, multiple 5-minute data are simulated and generated in this project. Longer period of data in terms of hours or days could be generated in order to model real life scenarios. The time intervals to slice the data should be adjusted accordingly. Furthermore, instead of simulating the movement of individuals with muscle stiffness, the models would be more robust given the real movement data of individuals with different walking gait. Last but not least, instead of classifying all types of anomalous data as "abnormal", ways to produce continuous value to quantify the degree of abnormality in both activity sequence and quality of movement context could be explored. As a starting point, the TCN model showed promising results based on the range of reconstruction error.

Bibliography

[1] Khaled A Alaghbari, Mohamad Hanif Md Saad, Aini Hussain, and Muhammad Raisul Alam. Activities recognition, anomaly detection and next activity prediction based on neural networks in smart homes. *IEEE Access*, 10:28219–28232, 2022.

[2] Raza Ali, Mohamed ElHelw, Louis Atallah, Benny Lo, and Guang-Zhong Yang. Pattern mining for routine behaviour discovery in pervasive healthcare environments. In *2008 International Conference on Information Technology and Applications in Biomedicine*, pages 241–244. IEEE, 2008.

[3] Damla Arifoglu and Abdelhamid Bouchachia. Activity recognition and abnormal behaviour detection with recurrent neural networks. *Procedia Computer Science*, 110:86–93, 2017.

[4] Damla Arifoglu and Abdelhamid Bouchachia. Detection of abnormal behaviour for dementia sufferers using convolutional neural networks. *Artificial intelligence in medicine*, 94:88–95, 2019.

[5] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[6] UABUA Bakar, Hemant Ghayvat, SF Hasanm, and Subhas Chandra Mukhopadhyay. Activity and anomaly detection in smart home: A survey. *Next Generation Sensors and Systems*, pages 191–220, 2015.

[7] Fabien Cardinaux, Simon Brownsell, Mark Hawley, and David Bradley. Modelling of behavioural patterns for abnormality detection in the context of lifestyle reassurance. In *Progress in Pattern Recognition, Image Analysis and Applications: 13th Iberoamerican Congress on Pattern Recognition, CIARP 2008, Havana, Cuba, September 9-12, 2008. Proceedings 13*, pages 243–251. Springer, 2008.

[8] Ayan Chatterjee and Bestoun S Ahmed. Iot anomaly detection methods and applications: A survey. *Internet of Things*, 19:100568, 2022.

[9] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012.

[10] Shirin Enshaeifar, Ahmed Zoha, Andreas Markides, Severin Skillman, Sahr Thomas Acton, Tarek Elsaleh, Masoud Hassanpour, Alireza Ahrabian, Mark Kenny, Stuart Klein, et al. Health management and pattern analysis of daily living activities of people with dementia using in-home sensors and machine learning techniques. *PloS one*, 13(5):e0195605, 2018.

[11] Shirin Enshaeifar, Ahmed Zoha, Severin Skillman, Andreas Markides, Sahr Thomas Acton, Tarek Elsaleh, Mark Kenny, Helen Rostill, Ramin Nilforooshan, and Payam Barnaghi. Machine learning methods for detecting urinary tract infection and analysing daily living activities in people with dementia. *PloS one*, 14(1):e0209909, 2019.

[12] Labiba Gillani Fahad and Syed Fahad Tahir. Activity recognition and anomaly detection in smart homes. *Neurocomputing*, 423:362–372, 2021.

[13] Abdur Rahim Mohammad Forkan, Ibrahim Khalil, Zahir Tari, Sebti Foufou, and Abdelaziz Bouras. A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. *Pattern Recognition*, 48(3):628–641, 2015.

[14] Guangyue Han and Brian Marcus. Analyticity of entropy rate of hidden markov chains. *IEEE Transactions on Information Theory*, 52(12):5251–5266, 2006.

[15] Yangdong He and Jiabao Zhao. Temporal convolutional networks for anomaly detection in time series. In *Journal of Physics: Conference Series*, volume 1213, page 042050. IOP Publishing, 2019.

[16] Ya-Xuan Hung, Chih-Yen Chiang, Steen J Hsu, and Chia-Tai Chan. Abnormality detection for improving elder's daily life independent. In *Aging Friendly Technology for Health and Independence: 8th International Conference on Smart Homes and Health Telematics, ICOST 2010, Seoul, Korea, June 22-24, 2010. Proceedings 8*, pages 186–194. Springer, 2010.

[17] Dorothy N Monekosso and Paolo Remagnino. Anomalous behavior detection: Supporting independent living. *Intelligent Environments*, pages 33–48, 2009.

[18] Marek Novák, Miroslav Biňas, and František Jakab. Unobtrusive anomaly detection in presence of elderly in a smart-home environment. In *2012 ELEKTRO*, pages 341–344. IEEE, 2012.

[19] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[20] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[21] Shao-Meng Qin, Hannu Verkasalo, Mikael Mohtaschemi, Tuomo Hartonen, and Mikko Alava. Patterns, entropy, and predictability of human mobility and life. *PloS one*, 7(12):e51353, 2012.

[22] Charissa Ann Ronao and Sung-Bae Cho. Deep convolutional neural networks for human activity recognition with smartphone sensors. In *Neural Information Processing: 22nd International Conference, ICONIP 2015, November 9-12, 2015, Proceedings, Part IV 22*, pages 46–53. Springer, 2015.

[23] Xiaobin Tan and Hongsheng Xi. Hidden semi-markov model for anomaly detection. *Applied Mathematics and Computation*, 205(2):562–567, 2008.

[24] Katerina Vrotsou, Kajsa Ellegard, and Matthew Cooper. Everyday life discoveries: Mining and visualizing activity patterns in social science diary data. In *2007 11th International Conference Information Visualization (IV'07)*, pages 130–138. IEEE, 2007.

[25] Kai Wang, Youjin Zhao, Qingyu Xiong, Min Fan, Guotan Sun, Longkun Ma, Tong Liu, et al. Research on healthy anomaly detection model based on deep learning from multiple time-series physiological signals. *Scientific Programming*, 2016, 2016.

[26] Zhiqiang Wang, Catherine Da Cunha, Mathieu Ritou, and Benoît Furet. Comparison of k-means and gmm methods for contextual clustering in hsm. *Procedia Manufacturing*, 28:154–159, 2019.

[27] Yan Zhou and Li Shang. Time sequence features extraction algorithm of lying speech based on sparse cnn and lstm. In *International Conference on Intelligent Computing*, pages 95–104. Springer, 2020.