VRIJE UNIVERSITEIT BRUSSEL

Graduation thesis submitted in partial fulfilment of the requirements for the degree of **Master of Science in Engineering: Computer Science**

PREDICTING ABNORMAL POSITIONS FOR WHEELCHAIRS

Detecting abnormal positions and falls of a wheelchair from real-time accelerometer data using OCSVM

Wacim Sid-Ali HALIT

January 2025

Promotor: Prof. Dr. Ann Nowé
Supervisor: Dr. Denis Steckelmacher

Faculty of Sciences and Bioengineering Sciences

VRIJE
UNIVERSITEIT
BRUSSEL

Proefschrift ingediend met het oog op het behalen van de graad van **Master of Science in de Ingenieurswetenschappen: Computerwetenschappen**

ABNORMALE POSITIES VOOR ROLSTOELEN VOORSPELLEN

Detecteren van Abnormale Posities en Valpartijen van een Rolstoel met Real-Time Versnellingsmetergegevens met behulp van OCSVM

Wacim Sid-Ali HALIT

Januari 2025

Promotor: Prof. Dr. Ann Nowé
Supervisor: Dr. Denis Steckelmacher

Faculteit Wetenschappen en Bio-ingenieurswetenschappen

Abstract

Wheelchair users face unique challenges in urban mobility, often navigating complex infrastructure with limited technological support for personal safety.

This thesis aims to design and implement a fall detection model using machine learning, integrated into a mobile application tailored to meet the specific needs of Belgian wheelchair users. The app will offer advanced mapping features, including filters for wheelchair-accessible cyclable and pedestrian routes, and incorporate gamification achievements to enhance user engagement towards the application.

The project is divided into four key phases: (1) the design and development of a mobile application with user-centric features, (2) the implementation of a threshold-based fall detection algorithm to establish a baseline, (3) the development and training of a One-Class Support Vector Machine (OCSVM) model and (4) a comprehensive comparative analysis to assess the effectiveness and reliability of both methods to determine which approach is more effective in accurately detecting falls and abnormal positioning of wheelchairs.

The methodology involves the creation of a custom dataset generated by simulating different scenarios of falls in the laboraty of the Vrije Universiteit Brussels, using a wheelchair equipped with specialized sensors.

Acknowledgements

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this thesis through their support, guidance, and expertise.

I am particularly thankful to Dr. Denis Steckelmacher for his invaluable supervision throughout this project. His expertise in the field, availability to answer questions, creative insights, and constructive feedback have been instrumental in shaping this research.

Special thanks go to Seyedeh Hiva Houshyar Yazdian for her significant contribution to the dataset creation and her continuous involvement throughout the project development.

I extend my appreciation to Tom and the AugmentX Lab team for providing access to their facilities for our fall simulation experiments, which were essential for the empirical aspects of this study.

Finally, I would like to thank my family and friends for their unwavering support, encouragement, and understanding throughout this academic journey. Their moral support has been a constant source of motivation and strength.

Contents

Chapter 1

Introduction

Disability is now an integral aspect of human experience. According to estimates from the World Health Organization, approximately 1.3 billion people (16% of the world's population) live with a significant disability ([World Health Organization, 2023]). Among these individuals are those who rely on wheelchairs. In developed countries, approximately 95% of people with disabilities have access to a wheelchair. In 34 developed nations, about 1% of the population, or about 10 million people, require wheelchairs, while in contrast, there are nearly 2% of the population, which is equivalent to approximately 121.8 million people in 156 developing countries ([Mobility, n.d.]). Wheelchairs users have often observed that their competence, intelligence, abilities, and overall experience with disability are frequently judged solely based on the visible presence of their wheelchair ([Saia et al., 2024]). In addition, they often face significant gaps in psychological support and are underrepresented in research efforts to understand and address their unique needs ([Furnham and Thompson, 1994]). However, improvements are not only limited to the psychological aspect; there is also a substantial need to enhance their safety and accessibility. A crucial area that requires special attention is facilitating access to new technologies ([Cooper et al., 2008]).

This thesis is part of a larger project aimed at creating an application fully tailored to the specific needs of Belgian wheelchair users. Therefore, we will design and develop the application with a strong emphasis on accessibility throughout the development phase, including feature implementation and interface design, to ensure a user-friendly experience for this target group [Ballantyne et al., 2018]. Our goal is to facilitate seamless integration of the application into users' daily routines, as poor usability not only affects user productivity but may also result in reduced long-term engagement, highlighting the crucial importance of an intuitive and accessible design [Shitkova et al., 2015]. The detection of falls and abnormal positions using sensors constitutes the central focus of this thesis and is arguably its most critical component. This feature plays an important role in ensuring the safety of wheelchair users by enabling real-time detection of falls, thus facilitating rapid intervention and assistance. In this study, we will compare the effectiveness of two methods: a threshold-based algorithm and an anomaly detection approach using a One-Class Support Vector Machine (OCSVM) [Libak Abou and Rice, 2023].

Chapter 2

Background

This chapter presents the essential knowledge required to understand our work and the concepts underlying the proposed methodology. It covers an explanation of the sensors utilized, the framework adopted for application development, OpenStreetMap (the API employed for mapping), and the concept of Bluetooth Low Energy (BLE), which serves as the communication medium between the phone and the sensors for data transmission. Furthermore, the theoretical foundations of the models applied in our use case, specifically Support Vector Machine (SVM) and One-Class Support Vector Machine (OCSVM) will be detailed.

2.1 Improving Wheelchair Systems Using Sensors

With the growing need for wheelchairs, numerous efforts have been made to improve their design and functionality. One example is the development of a load-cell-based wheelchair stability assessment system (Wheel-SAS) [Moody, 2012], which was designed to improve the chair's stability, providing users with greater comfort and reducing potential disturbances during use. Laboratory tests and experimental trials have also contributed to the development of advanced transducers and control systems for powered wheelchairs as these innovations focused on enhancing maneuverability, minimizing veering on inclined surfaces, improving energy efficiency, and reducing the physical effort required from users [Sanders et al., 2010].

This review of the literature highlights the significant efforts dedicated to enhancing the comfort and safety of wheelchair users by improving both the design and functionality of wheelchairs. Various sensor-based systems have been integrated into wheelchairs, enabling features such as autonomous navigation to assist individuals with disabilities in their daily mobility [Schilling et al., 1998]. It appears that the primary focus has been on proactive measures aimed at preventing undesirable incidents, such as veering off paths or navigating obstacles. However, a noticeable gap remains in addressing strategies to help wheelchair users after an incident has already occurred, such as a fall or abnormal positioning.

Knowing this, The focus of this thesis was put on the development and implementation of a sensor-based fall detection system (detailed in the next section) in order to provide assistance to users after the incidents occur.

2.1.1 Sensors Utilized In The Project

Having reviewed the state-of-the-art developments regarding sensors used in wheelchairs, we will now delve into the specific sensors employed in our project. This section builds upon the prior thesis work of a colleague Abdulaziz DOUKHAN, and the descriptions and images used to illustrate the functionality of these sensors are derived from his work and contribution ([Abdulaziz, 2024]).

ESP32S3 Microcontroller

The ESP32S3 microcontroller, developed by Espressif Systems, was selected due to its compact size, high performance, and cost-effectiveness ([Plauska et al., 2022]). In the photo below, it is shown connected to a camera and a microSD card, reflecting our setup.

Figure 2.1: Photo Of The ESP32S3 Microcontroller

Absolute Orientation Sensor - BNO055

This sensor is utilized for precise orientation and motion tracking. It was used to monitor wheelchair orientation and speed at specific timestamps, providing three values that correspond to the X, Y, and Z axes. Its choice selection is justified by its high level of accuracy.

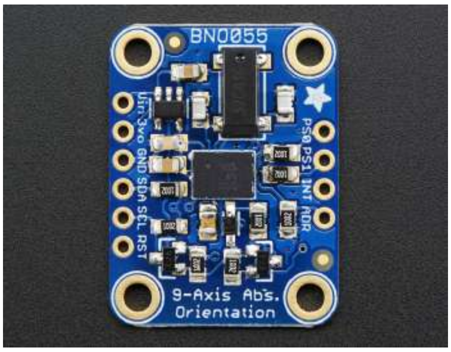


Figure 2.2: Photo Of The BNO055 Sensor

Humidity sensor - BME280

Although this sensor was not used in our specific contribution, it is worth mentioning its relevance to the final application. It will offer valuable information about weather conditions, which could potentially be used to suggest routes and adapt them based on the weather.

Figure 2.3: Photo Of The BME280 Sensor

CR-18650 lithium-ion battery

To ensure optimal performance and durability when connected to a device—along with the use of Bluetooth Low Energy (which we will discuss in the next section) a CR-18650 lithium-ion battery has been utilized.

Figure 2.4: Photo Of The BME280 Sensor

Sensor Box

All of these components were assembled and enclosed in a box designed and 3D-printed by Dr. Denis STECKELMACHER (See Figure 2.5).

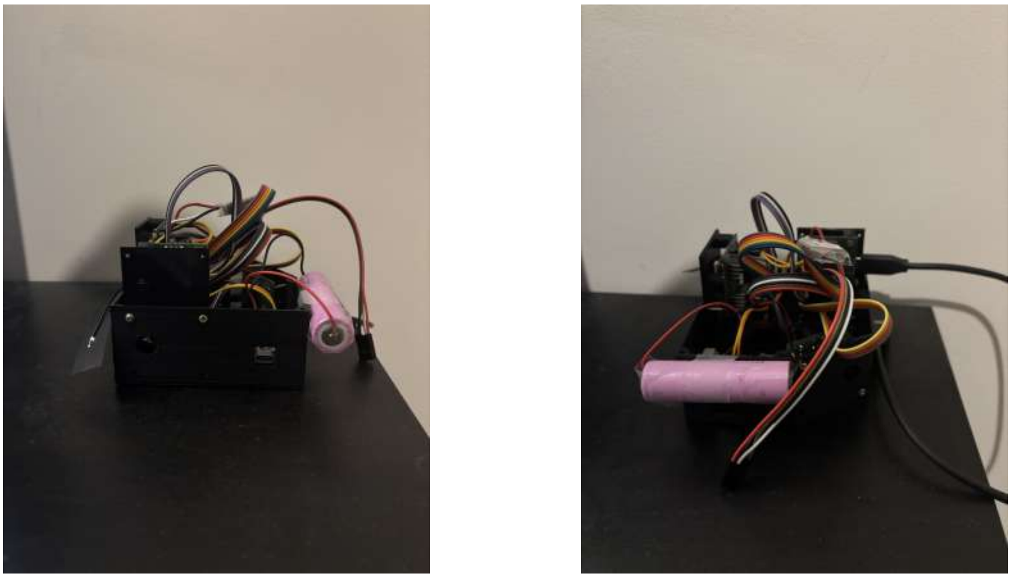


Figure 2.5: Photos Of The Sensor's Box (front and side)

2.2 Bluetooth Low Energy

Before diving into Bluetooth Low Energy (BLE), it is essential to first understand standard Bluetooth technology. Bluetooth is a widely used wireless communication protocol designed for short-range data exchange between devices such as smartphones, computers, and wireless peripherals. This technology eliminates the need for physical cables and allows quick and convenient connections between devices ([ABET, 2021]). Although its potential was evident early on, traditional Bluetooth faced limitations, especially in applications requiring low power consumption and long-term use.

To address these challenges, Bluetooth Low Energy (BLE) was introduced as a more efficient alternative. BLE complements standard Bluetooth by offering significantly lower energy consumption while maintaining effective wireless communication ([Heydon and Hunn, 2012]). As its name implies, BLE prioritizes energy efficiency, making it an adequate solution for our case, which focuses on creating an assistive application to help wheelchair users with their daily activities, hence reducing battery consumption was an important factor to take into account. Ensuring that the application does not excessively drain the phone's battery allows users to maximize its usability throughout the day. Moreover, BLE supports the simultaneous connection of multiple devices, which is crucial for our application, as it depends on the continuous data transmission from various sensors. Another advantage is Bluetooth Low Energy's ability to facilitate fast and automatic device pairing, simplifying the user experience and ensuring seamless connectivity. These characteristics has made BLE a dependable and efficient choice for meeting the specific requirements of our assistive application.

The Figure 2.6 depicts the workflow of a Bluetooth Low Energy system that involves the interaction between a BLE device, smart devices and a cloud server.
Here is an explanation of each step:

1. Nearby smart devices (like smartphones) equipped with Bluetooth Low Energy technology detect these radio signals.

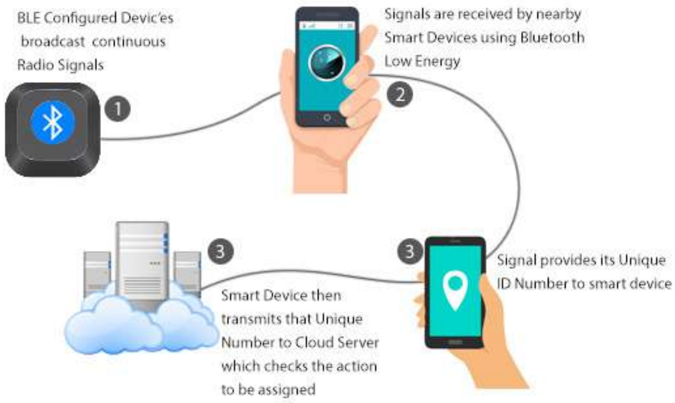


Figure 2.6: Workflow Of Bluetooth Low Energy. Source : https://www.einfochips.com/blog/bluetooth-low-energy-ble-the-future-of-retail-technologies/

2. The received signal provides the smart device with a unique ID number associated with the BLE device.

3. The smart device transmits the unique ID number from the BLE device to a cloud server.

4. The cloud server processes this information and determines the appropriate action to be performed based on the received data.

2.3 Open Street Map

OpenStreetMap, originally founded by Steve COAST in 2004, was created to develop an open source geographic database of the world ([Bennett, 2010]). Although initially focused on mapping streets, it has since expanded to include features such as footpaths, buildings, cycling routes, pedestrian pathways, and bridges.

We decided to use OpenStreetMap instead of the Google Maps API, even though Google Maps is often seen as one of the most popular mapping services ([Mehta et al., 2019]). The main reason for this choice is the open source nature of OpenStreetMap, which fits perfectly with the goals of our project. It provides free access to a wide range of maps, features, and filters, making it easier to adapt and customize for the specific needs of our use case. This flexibility allowed us to create tailored functionalities, such as applying specific filters to pedestrian and bicycle roads.

Among the various maps available, there is one specifically dedicated to Belgium. OpenStreetMap Belgium has been actively promoting the OpenStreetMap project by co-organizing events such as:

- FOSS4G Belgium

- The Annual Open Belgium Conference

Figure 2.7: Screenshot Of The Location Of The Vrije Universiteit Brussels in openStreetMap.

2.4 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a robust machine learning algorithm commonly used for linear and nonlinear classification, regression, and outlier detection. It works by finding a hyperplane that best separates different classes of data. The main goal is to maximize the margin between the data points of different classes while ensuring that the data points are correctly classified ([GeeksforGeeks]).

Key Components

The SVM algorithm relies on several fundamental components:

- **Hyperplane:** In an n-dimensional space, a hyperplane is an $(n-1)$-dimensional flat affine subspace that divides the space into two halves. For example, in 2D, a hyperplane is a line, and in 3D it is a plane, as illustrated in Figure 2.9.
- **Margin:** The margin is the distance between the hyperplane and the nearest data points of each class. These nearest points are called support vectors.
- **Support Vectors:** These are the data points that lie closest to the hyperplane and define the margin. They are crucial as they are the only data points used to determine the optimal hyperplane.

Mathematical Foundation

The SVM classifier can be mathematically expressed as:

$$f(x) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b) \tag{2.1}$$

Where:

- $K(x_i, x)$ represents the kernel function

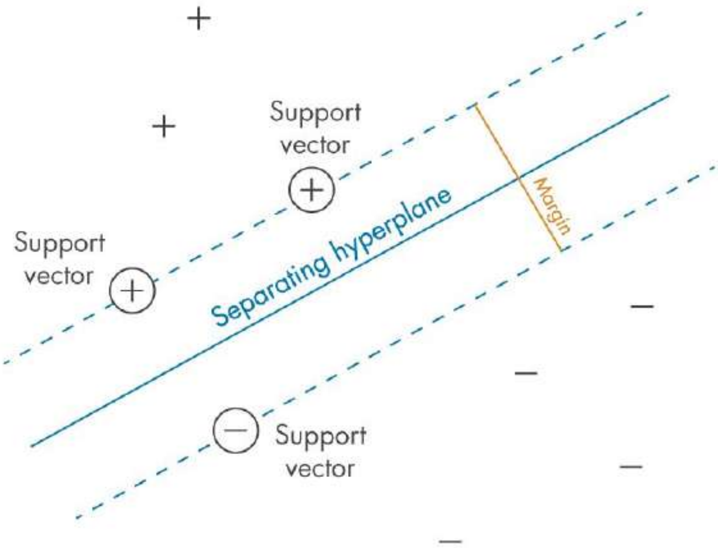


Figure 2.8: Visualization of a Support Vector Machine (SVM) demonstrating the optimal hyperplane and support vectors.

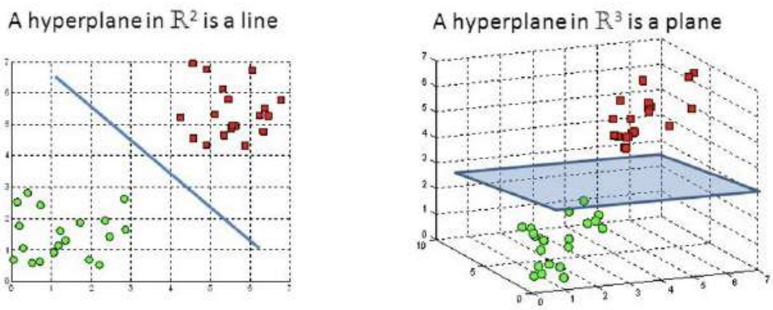


Figure 2.9: Hyperplanes in 2D and 3D feature space, illustrating how the decision boundary changes with dimensionality.

- α_i are the Lagrange multipliers
- y_i are the class labels
- b is the bias term

Kernel Functions

To handle non-linearly separable data, SVMs employ kernel functions that map the data to a higher-dimensional space where linear separation becomes possible. Common kernel functions include:

- Linear: $K(x_i, x_j) = x_i^T x_j$
- RBF (Gaussian): $K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$

Optimization Problem

Finding the optimal hyperplane involves solving a quadratic optimization problem:

1. The objective is to maximize $\frac{2}{\|\mathbf{w}\|}$, where $\mathbf{w}$ is the normal weight vector to the hyperplane.
2. The constraints ensure correct classification of data points:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \tag{2.2}$$

where $\mathbf{x}_i$ are the feature vectors, y_i are the class labels ($+1$ or -1), and b is the bias term.

2.5 Traditional SVM vs One-Class SVM

While traditional SVMs focus on finding a boundary between two or more classes, One-Class SVM (OCSVM) addresses a fundamentally different problem: identifying anomalies or outliers in a dataset where only normal instances are available for training. The key differences include:

- **Training Data**: Traditional SVMs require labeled data from multiple classes, while OCSVM learns from a single class which consists of normal instances.
- **Decision Boundary**: OCSVM creates a boundary that encompasses the normal data points, treating everything outside as anomalies.
- **Objective Function**: Rather than maximizing the margin between classes, OCSVM aims to find the smallest hypersphere that contains most of the normal data

In the context of this thesis, the primary goal is to detect rare events, such as falls or abnormal orientations of the wheelchair. Given that our dataset (which is discussed in detail in a later chapter) contains a significantly higher number of normal instances compared to falls, this class imbalance has influenced the choice of using this model. OCSVM presents several advantages that make it particularly appropriate for our specific use case:

- The algorithm can be trained exclusively on normal wheelchair movement patterns, eliminating the need for a large collection of fall samples.

- The method makes no assumptions about the underlying data distribution of wheelchair movements, providing flexibility in detecting various fall patterns
- Through the ν parameter, we can control the model's sensitivity to potential anomalies in the training data, ensuring robust learning of normal movement patterns

Chapter 3

Related Work

In this chapter, we will explore existing research and projects in the field of fall detection systems. We will examine studies that use threshold-based algorithms, specifically those that take advantage of sensor data, to detect falls. This review will also include a comparison of threshold-based methods with machine learning approaches, with a particular focus on the application of Support Vector Machines (SVMs) for fall detection.

3.1 Fall Detection

Using wheelchairs for daily mobility can increase the risk of falls due to difficulties maneuvering the wheels or external factors, such as slippery surfaces, that can compromise stability. These falls are a well-known source of mortality in the elderly ([Sterling et al., 2001]). Traditional wheelchairs often demand assistance or use of more physical effort than usual. This extra effort and challenges faced by individuals with disabilities have pushed technological advancements, leading to the creation of more user-friendly support and monitoring systems for patients [Rahman et al., 2020]. These innovations have gained increasing prominence due to their impact on improving the daily lives of wheelchair users.

Fall detection systems are a widely researched aspect of Ambient Assisted Living (AAL) applications aimed at ensuring safety. Despite the variety of approaches proposed for fall detection, there remains a special need for precise and reliable architectures, methods, and protocols tailored specifically to wheelchair users ([Sheikh and Jilani, 2023]). This is why we seek to conduct a comparison between two approaches: the threshold algorithm method and the AI-based method, with the goal of evaluating their performance and identifying which one delivers more accurate results for fall detection and abnormal positioning of the wheel chair after the fall has occured.

In our use case, the sensor box will be placed directly on the wheelchair, representing a novel approach compared to previous works that commonly placed the acceleration sensor on the subject's chest due to its proximity to the body's center of gravity ([Lim et al., 2014]). Similarly, while many projects rely on smartphones for fall detection, this method has a notable limitation: smartphones are not always consistently positioned on the body, which can compromise detection accuracy or even disable the functionality entirely ([Tao Xu and Liu, 2021]).

Figure 3.1 is extracted from the work of Sara Usmani et al. ([Usmani et al., 2021]) and illustrates the sensor placements explored in recent research efforts. It highlights the various locations used in studies to optimize sensor-based data collection for activity monitoring and fall detection. We can see that the sensors in prior research have predominantly been positioned on the individual's body rather than on the wheelchair itself. This highlights the novelty of the proposed positioning in our approach. By fixing the sensor box to a predefined position on the wheelchair, we ensure consistent data acquisition and allow for precise computation of values and thresholds, leading to more accurate and reliable fall predictions and further data analysis.

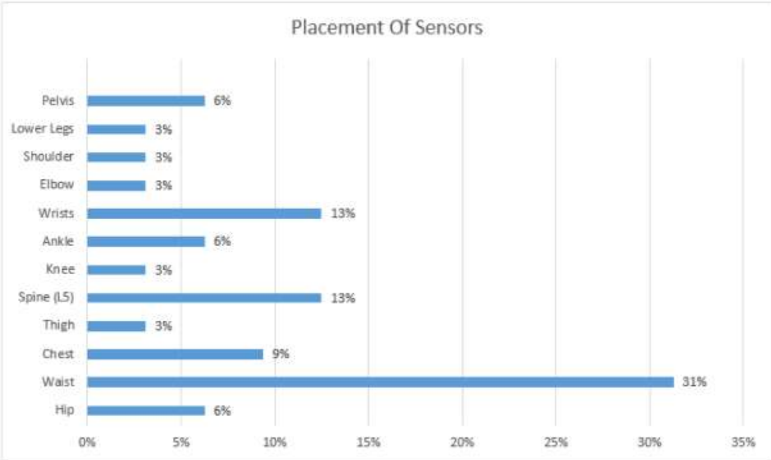


Figure 3.1: Placement Of Sensors

As illustrated in Figure 3.2, extracted also from ([Usmani et al., 2021]), the figure highlights the most commonly utilized machine learning algorithms in recent research. We can see that the SVM is the most present with 29%.

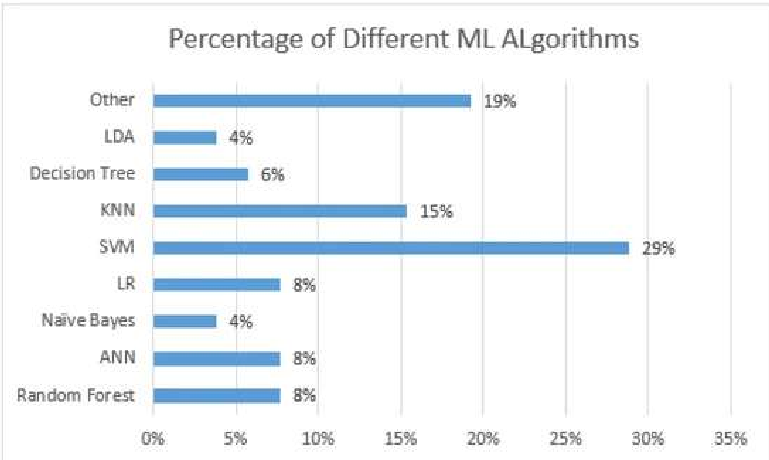


Figure 3.2: Frequency of Usage Of The Different Machine Learning Algorithms.

3.1.1 Treshold-based Algorithms

Sensors have been widely used to monitor human activity. In threshold-based algorithms, the features are derived from measured accelerations and assessed using a set of rules to determine if a fall has occurred or not, making the choice of fixed thresholds essential for the effectiveness and accuracy of the algorithm ([Razum et al., 2018]).

In their work, Bourke, O'Brien, and Lyons employed tri-axial accelerometer sensors for fall detection ([Bourke et al., 2007]). They placed the sensors on the trunk and thigh of individuals, and the resultant signal from both was derived by taking the root-sum-of-squares of the three signals from each tri-axial accelerometer recording. When stationary, it would remain constant at (+1 g) (which is approximately 9.81, corresponding to the acceleration due to gravity).

Four thresholds were computed:$\mathrm{UFT_{TRUNK}}, \mathrm{LFT_{TRUNK}}, \mathrm{UFT_{THIGH}}, and \mathrm{LFT_{THIGH}}$.

as shown in Figure 3.3 present in their paper. A representative signal for a fall, along with recorded fall and ADL signals, is displayed.

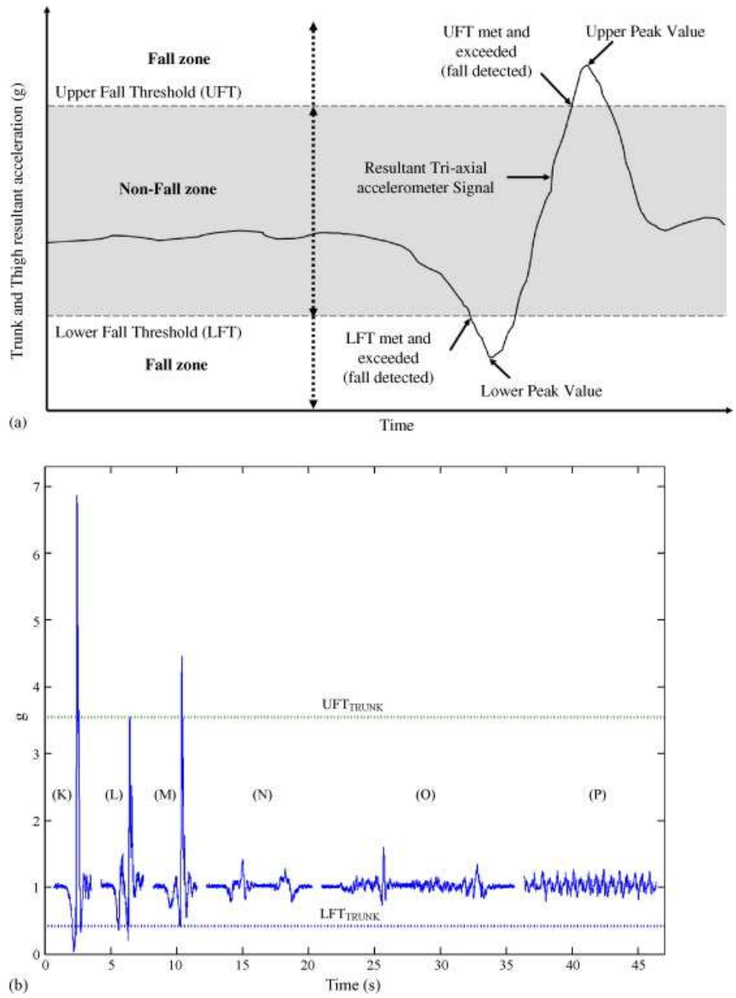


Figure 3.3: Fig. 1. Fall detection algorithm operation example for upper and lower thresholds, using an artificial example signal (a). Trunk resultant vector signals (b) for a typical fall (K), the fall that produced the smallest magnitude UPV (L), the fall that produced the smallest magnitude LPV (M), a typical sitting on an armchair activity (N), a getting in and out of a car seat activity (O) and walking (P). Source : https://www.sciencedirect.com/science/article/pii/S0966636206001895

The system's concept is illustrated through two key graphs: a theoretical model and actual experimental data. In the theoretical model, a "Non-Fall zone" is established between an Upper Fall Threshold (UFT) and Lower Fall Threshold (LFT), with any acceleration signals exceeding these boundaries indicating a fall event. During normal activity, the resultant acceleration signal remains relatively stable around 1g (approximately 9.81 m/s^2), representing the constant pull of gravity. However, during a fall, the signal exhibits distinctive patterns, including pronounced peaks that breach either the upper or lower thresholds. The experimental data graph demonstrates this in practice, showing several significant spikes in acceleration around the 5s and 10s marks, clearly exceeding the established UFTtrunk and LFTtrunk thresholds. These spikes represent potential fall events, distinguishing them from normal movement patterns.

3.1.2 Use Of Support Vector Machine

A significant portion of the literature on the use of Support Vector Machines (SVMs) for the prediction of falls, particularly for elderly or wheelchair users, has been heavily based on postural analysis and information derived from image data ([Foroughi et al., 2008]).

One approach explored in this regard involves using a single camera to monitor the entire room environment, capturing video recordings of an elderly person's daily activities over a specific time period. Features are then extracted from these recordings, and an online one-class Support Vector Machine (OCSVM) is applied to identify regions in the feature space that differentiate between normal daily postures and abnormal postures, such as falls ([Yu et al., 2013]).

Chapter 4

Overview Of The Contribution

This chapter describes the methodology employed in this thesis, providing a detailed account of the application's development and the functionalities implemented. It also covers the process of dataset creation, feature engineering, and fall detection implementation.

This thesis being a part of a larger project with the end goal of developing an application that:

1. Provides optimal routing specifically designed for wheelchair users (a discussion on this aspect will follow).
2. Real-time emotion detection system
3. Fall detection and alert system
4. Achievement tracking to encourage user engagement
5. Personalized user experience based on wheelchair proficiency level
6. Bluetooth integration for sensor data collection

Research Focus Our primary contribution will focus on developing and implementing an optimal fall detection method. Although our initial implementation will serve as a foundation, it will be designed to accommodate future enhancements and extensions.

User Interface and Features The design and development of the application will also be addressed, featuring a comprehensive user interface that includes an interactive map system integrated with sensor connectivity. Additionally, we will implement specialized map filters to highlight wheelchair-accessible routes, such as bicycle paths and pedestrian walkways. This will allow users to easily identify the most suitable routes.

Proposed Additional Features The application will include:

- A dedicated Help section with wheelchair usage guidelines.
- Multilingual support in Dutch, English, and French, reflecting the predominant languages spoken in Brussels based on local demographic observations.

4.1 Design and Development of the Application

We begin this chapter by discussing the design and development process of the application, as it was the initial step of this project and also its first significant challenge.

The first consideration that came to mind when conceptualizing this application was addressing the question "What should the application look like?". Finding the right balance between user experience and a clean, simple design was essential for the application's development. Studies in human-computer interaction (HCI) have underscored the crucial importance of user experience in the success of assistive technologies. An important consideration is adopting an inclusive design approach, which emphasizes designing and validating solutions directly with end-users to ensure accessibility and usability ([Clarkson and Coleman, 2010]). Therefore, it is important that the application we develop is intuitive and easy to use, aligning with the principles of user-centered design to ensure a positive and accessible user experience ([Interaction Design Foundation - IxDF, 2016]).

4.1.1 Inspirations For The Design

After determining the features to be implemented, we needed to consider the most effective way to introduce them without making the process of discovery difficult for users. It was important to keep in mind that our goal is to simplify their daily lives and not add unnecessary challenges. Since our application required map integration, we took as an inspiration various available options. We opted for Floya, an application that offers comprehensive mobility services in Brussels, enabling users to travel wherever, whenever, and however they want. In addition, we saw Plan, the iOS-provided routing application.

As shown in Figure 4.1, the screenshots illustrate the main pages of the Floya and Plan applications.

(a) Screenshot of the application Floya

(b) Screenshot of the application Plan (IOS)

Figure 4.1: Screenshot showing the main pages of the Floya and Plan applications

4.1.2 Our Application Design Proposal

The following section presents screenshots of each feature discussed previously, accompanied by detailed functional explanations.

4.1.3 Questions about general use of the wheelchair

It was mentioned before that the final application aims to provide personalized route recommendations based on the proficiency of individual users in wheelchairs and their personal approach to mobility challenges. To achieve this customization, we implemented a four-page questionnaire in which users can select from three options for each question, following a Likert scale-like format. User responses are stored in JSON format, serving dual purposes: personalizing the user experience and collecting data for future research and analysis.

It should be noted that the current questionnaire structure was developed based on theoretical and personal understanding of the problem, without direct input from wheelchair users. As such, these assessment criteria are preliminary and flexible, designed to be refined through future user feedback and potential empirical validation (See Figure 3.2).

4.1.4 Homepage

The application maintains a consistent layout across all pages, having a dedicated content area for page-specific features and a fixed bottom navigation bar for inter-page navigation.

Upon launch, users are presented with a home page containing an interactive map interface that immediately pinpoints their location in real time.

Located at the top of the screen, the **'Cyclable'** and **'Walk'** buttons enable users to dynamically filter the map display through API calls, highlighting their respective route types. Two functional buttons are positioned at the bottom of the interface: **'Connect'** for establishing Bluetooth Low Energy device connections, and **'Start'** for beginning route navigation (See Figure 4.3).

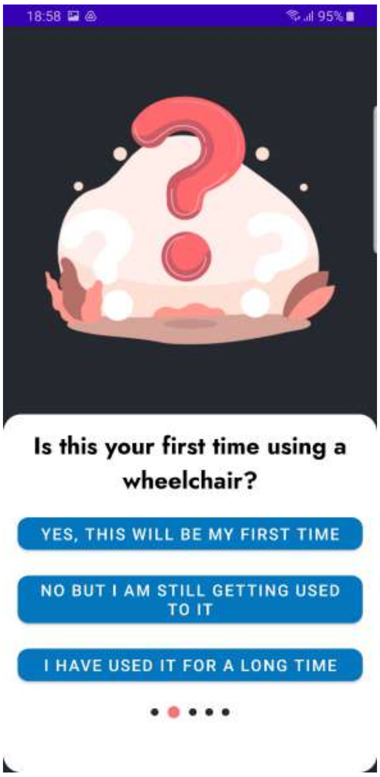
4.1.5 Achievement tracking

A gamification approach was proposed through an achievement system that would reward users based on their application usage and mobility metrics. This system would include milestones for criteria such as cumulative application usage time and total distance traveled in the wheelchair. While the implementation of this tracking system could not be realized due to the routing functionality being outside the scope of this project phase, we developed preliminary design mockups to establish the design foundation for the future (See Figure 4.4).

4.2 Challenges Of The App Development

The development of the Android application presented significant technical challenges, particularly in managing system permissions and hardware access. Two critical areas required careful implementation:

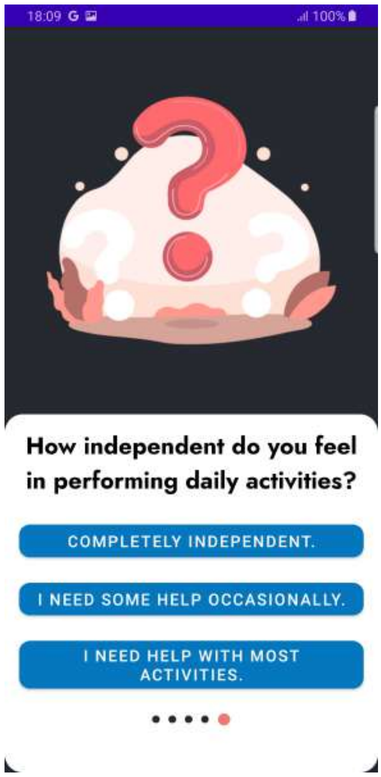
- Location services: Proper configuration of location permissions was essential for integrating the OpenStreetMap API, enabling accurate navigation functionality.



(a) Question 01

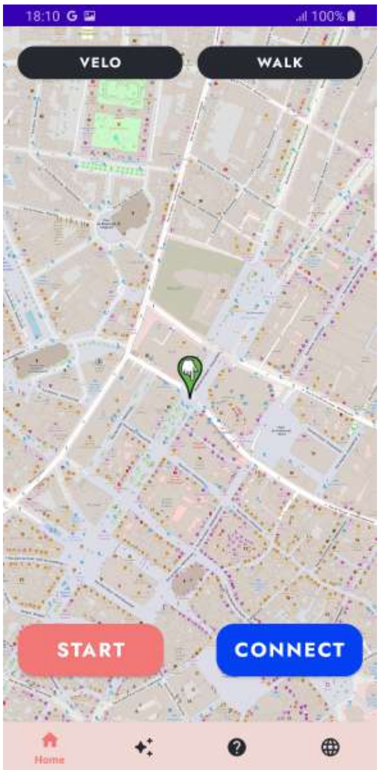
(b) Question 02

(c) Question 03



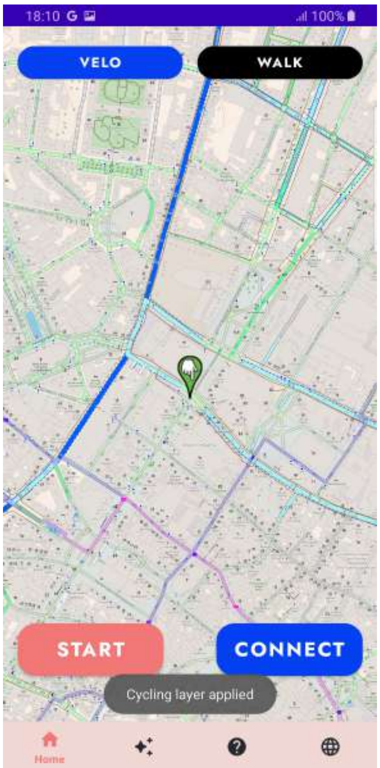
(d) Question 04

Figure 4.2: Survey interface of the application displaying questions designed to gather user feedback and preferences

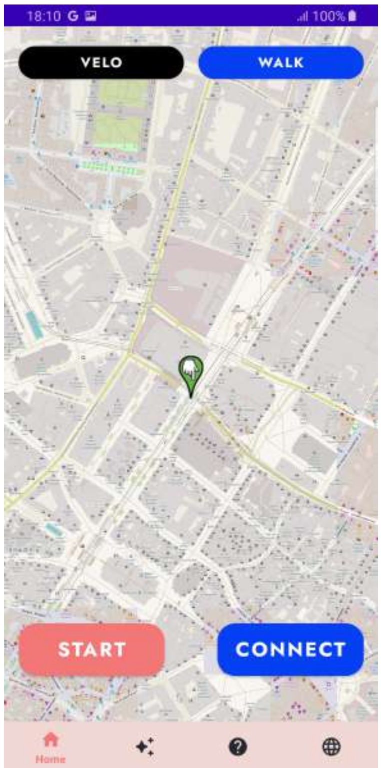


(a) Home Page (No Filter Applied)

(b) List Of BLE Device Detected

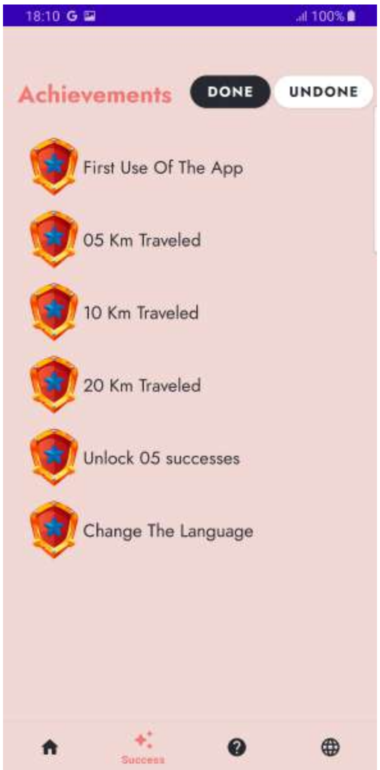


(c) Cyclable Routes Filter Applied



(d) Pedestrian Routes Filter Applied

Figure 4.3: Key components of the application's home page and its different functionalities



(a) Achievements page (Done Filter Applied)

(b) Achievements page (Undone Filter Applied)

(c) Achievements page (No Filter Applied)

Figure 4.4: Screenshot Of Achievement Tracking

- Bluetooth connectivity: Implementation of Bluetooth Low Energy (BLE) device detection required specific runtime permissions.

These requirements necessitated thorough understanding of Android's permission system and security model to ensure proper functionality while maintaining user privacy and system security.

4.3 Data Collection Methodology

During the data collection phase, we encountered several initial challenges in sensor box positioning. The preliminary design was supposed to place the box on the side of a wheelchair using a ruler-like support, which was not yet available. Consequently, we relied solely on strings for attachment, resulting in a little unstable sensor box placement. To mitigate potential damage during fall simulations, we strategically positioned the box at the rear of the wheelchair, as illustrated in Figure 4.5.

We conducted a comprehensive total of 39 recording sessions at the Augmex Experiment Lab of the Vrije Universiteit Brussels, which provided us with the necessary infrastructure to simulate falls and incorporate various do-it-yourself obstacles. All sessions were recorded to facilitate data interpretation. The data collection sessions were organized as follows:

- **Sessions 1–7**: Initial sensor functionality testing
- **Sessions 8–19**: Normal wheelchair use (expanded to provide a larger training dataset for the Support Vector Machine (SVM) to enhance anomaly detection)
- **Sessions 20–24**: Backward falls (personally simulated with careful precautions, guided by online tutorial safety protocols)
- **Sessions 25–28**: Forward falls (conducted without a rider for safety, involving wheelchair propulsion and simultaneous throwing)
- **Sessions 29–37**: Side falls both in right and left side.
- **Sessions 38–39**: Normal walking on obstacles

This structured approach allowed us to comprehensively capture various movement scenarios and potential fall dynamics.

The protocol for the data collection was proposed by Hiva who has helped during all the data collection process. It is as follows :

1. **Objective**
    - To develop a reliable method for detecting falls and obstacles in wheelchairs using an Inertial Measurement Unit (IMU).
2. **Materials and Equipment**
    - IMU sensor
    - Wheelchair
    - Padded floor or gym mats
    - High-speed cameras (for visual analysis)
    - Data logging device (laptop, microcontroller connected to the IMU)

3. **Setup**
   - **IMU Placement:**
     - Attach the IMU securely to the wheelchair frame, preferably near the center of mass or another stable location.
     - Ensure the sensor orientation is aligned for consistent axis measurements.
   - **Environment Preparation:**
     - Use a flat, padded testing area to minimize impact damage.
     - Clear the area of obstacles to avoid interference during testing.
   - **Data Logging:**
     - Connect the IMU to the data logging device.
4. **Experimental Procedure**
   - **Step 1: Initial Baseline Recording**
     - Record baseline IMU data for static and normal wheelchair movement.
     - Ensure no noise or signal distortion is present in the data.
   - **Step 2: Controlled Fall Simulation**
     - Dynamic Backward Falls
     - Dynamic Forward Falls
     - Dynamic Side Falls (without user in wheelchair)
5. **Data Recording and Synchronization**
   - Ensure continuous recording of IMU data during all experiments.
   - Synchronize video timestamps with IMU logs.
   - Label all trials clearly.
6. **Data Analysis**
   - Preprocessing
   - Feature Extraction
   - Event Detection

4.4 Feature Engineering

The dataset contains the following features :

Environmental Features

1. `temperature`: Represents the ambient temperature during data collection.
2. `humidity`: Reflects the ambient humidity level, likely as a percentage.
3. `pressure`: Atmospheric pressure.
4. `altitude`: Indicates elevation changes.

(a) Back view of the wheelchair

(b) Front view of the wheelchair

(c) Position of the sensors box

(d) Side view of the wheelchair

Figure 4.5: Different view of the wheelchair used for data collection

Motion Features

5. `absolute_orientation[0]`, `absolute_orientation[1]`, `absolute_orientation[2]`: Orientation data in 3D space. Useful for determining the wheelchair's tilt, roll, or overall spatial orientation.

6. `angular_velocity[0]`, `angular_velocity[1]`, `angular_velocity[2]`: The rotational velocity along the X, Y, and Z axes.

7. `linear_acceleration[0]`, `linear_acceleration[1]`, `linear_acceleration[2]`: Linear acceleration values along the X, Y, and Z axes, measured in meters per second squared (m/s^2).

From these features, we derived additional ones to accurately compute moments of change in acceleration and orientation, allowing us to detect falls effectively. The formulas are fundamental principles of physics and mathematics; we simply manipulated the existing values to calculate new ones.

1. Magnitude of Acceleration

- **Definition**: Combines the X, Y, and Z components of linear acceleration into a single value representing the overall acceleration magnitude.

$$magnitude\_acceleration = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

- **Purpose**: Useful for detecting sudden changes in motion during a fall.

2. Magnitude of Angular Velocity

- **Definition**: Combines the X, Y, and Z components of angular velocity into a single value representing the overall rotational speed.

$$magnitude\_angular\_velocity = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$$

- **Purpose**: Useful for detecting rapid spins or tipping movements during a fall.

3. Tilt Angles: Computes the tilt of the wheelchair in terms of:

1. **Pitch**: Rotation around the X-axis.

$$pitch = \arctan 2(a_y, \sqrt{a_x^2 + a_z^2})$$

2. **Roll**: Rotation around the Y-axis.

$$roll = \arctan 2(-a_x, a_z)$$

3. **Derivation**: Based on the absolute orientation values, typically using trigonometric functions.

4. **Purpose**: Indicates if the wheelchair has tilted beyond a safe threshold, suggesting a fall.

The tilt angles are crucial for detecting whether the wheelchair user has fallen horizontally or sideways.

Tilted Sideways Detection

$$\text{Sideways Tilt} = |\theta_{\text{roll}}| > \theta_{\text{threshold}}$$

Threshold for detecting a sideways tilt (typically 45-60 degrees).

Tilted Forward/Backward Detection

$$\text{Forward/Backward Tilt} = |\theta_{\text{pitch}}| > \theta_{\text{threshold}}$$

Threshold for detecting forward/backward tilt (typically 45-60 degrees).

Completely Horizontal Detection

$$\text{Horizontal} = (|\theta_{\text{roll}}| < \delta) \wedge (|\theta_{\text{pitch}}| < \delta)$$

Small angle threshold (typically 10-15 degrees). This detection is useful in confirming that the box is in its intended position and stable. It can help ensure that small movements or vibrations do not result in unnecessary alerts. This can also be used as a baseline check before triggering other actions.

Note : While abnormal wheelchair positions can persist after a fall, our primary focus remains on the initial fall detection. Therefore, both the threshold-based algorithm and the One-Class Support Vector Machine model are optimized specifically to identify the onset of falls, as these represent the critical transition from normal to abnormal wheelchair states.

4.5 Threshold Algorithm

The fall detection algorithm follows a multi-step process to accurately identify fall events:

1. First, the system continuously monitors the magnitude of acceleration from the sensor data.
2. When this acceleration exceeds a predefined threshold, the algorithm proceeds to the next phase.
3. The system then implements a delay period, waiting until the large acceleration event concludes.
4. Following this delay, the algorithm computes the jerk (rate of change of acceleration) between:
    - The moment when the threshold was exceeded
    - The preceding moment before the threshold breach
5. If this computed jerk exceeds another predetermined threshold, the algorithm moves to the orientation analysis phase.
6. The orientation analysis examines two critical angles:
    - Roll angle
    - Pitch angle
7. If either the roll or pitch angle exceeds their respective thresholds, the system declares a fall detection event.

This approach ensures that the algorithm:

- Minimizes false positives by requiring multiple conditions to be met
- Considers both dynamic (acceleration, jerk) and static (orientation) features
- Implements temporal analysis through the delay period
- Validates the fall through final posture analysis

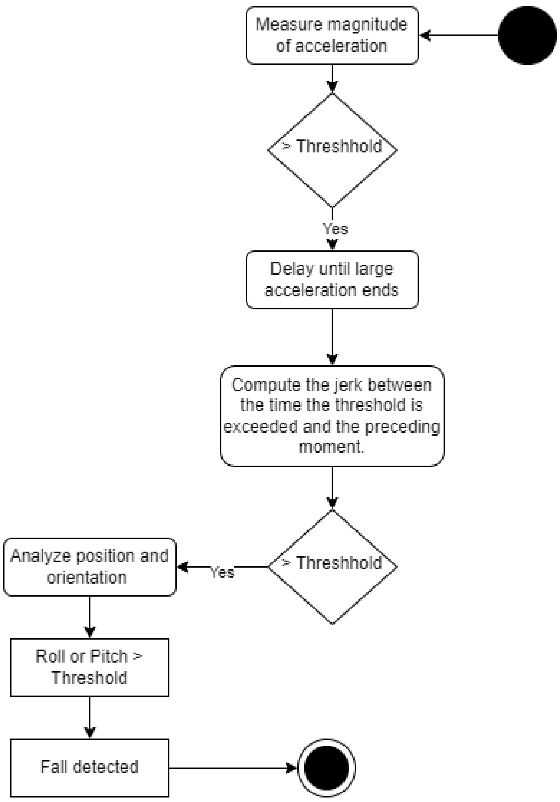


Figure 4.6: Flowchart of the proposed fall detection algorithm

The algorithm can be mathematically represented as:

$$FallDetected = \begin{cases} 1 & \text{if } (a > Th_a) \wedge (j > Th_j) \wedge (|\theta_{roll}| > Th_r \vee |\theta_{pitch}| > Th_p) \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where:

- a is the acceleration magnitude
- j is the computed jerk
- θ_{roll} and θ_{pitch} are the roll and pitch angles
- Th_a, Th_j, Th_r, and Th_p are their respective thresholds

Figure 4.6 illustrates the algorithmic workflow of our threshold-based fall detection system. The flowchart demonstrates the sequential decision process, beginning with acceleration measurement and progressing through multiple validation stages to ensure accurate fall detection.

4.6 One Class Support Vector Machine Model

During the development of our fall detection model, creating an appropriate dataset proved to be our main challenge. After calculating the key features like acceleration magnitude and jerk, we needed to properly label our data to indicate whether each instance was a fall or not.

To build this dataset, we first used the video recordings to identify the exact start time of each training session, making sure to exclude any initialization activities like wheel tapping. Then, we added a binary 'fall' feature as our target variable. Since we're working with anomaly detection, we needed our training set to contain only normal instances - this meant setting all 'fall' values to 0 in the training data. Data quality was essential for our approach. We had to be particularly careful to exclude any data that could be unclear or confusing for the model. This strict filtering was necessary because in anomaly detection, including any questionable data in the training set could seriously impact how well the model distinguishes between normal wheelchair use and actual falls. For our testing dataset, we incorporated various types of fall scenarios, including backward, forward, and lateral falls. Using video recordings as reference, we tried to carefully identify and delimite the time intervals during which each fall occurred. All data instances within these identified fall periods were labeled with a value of 1, as these represent the anomalous events our model aims to detect.

Chapter 5

Results

In this chapter, we present our experimental results, covering the data collection from wheelchair sensors, the extraction of key features, the implementation of detection thresholds, and our anomaly detection methodology for identifying fall events.

5.1 Interpretation Of The Data Collected

In this section, we will present a comprehensive visualization and analysis of the sensor data collected across the various experimental scenarios. Our analysis will provide an in-depth interpretation of the graphical representations, focusing on the distinctive spikes, curves, and feature characteristics. By correlating the sensor data with the synchronized video recordings, we will precisely contextualize the timing and nature of events for each experimental condition, including normal wheelchair use, backward falls, forward falls, side falls, and obstacle interactions.

Note : To identify the precise starting point for our data analysis, we deliberately tap the wheel to create a distinct spike in sensor readings. This tap creates a clear marker in the data that serves as our reference point. Once we identify this spike, we can crop the data starting from that point, effectively removing any irrelevant sensor readings that were recorded between when the sensors were turned on and when the actual experiment began. In the following graph, we will keep this initial spike visible to demonstrate this reference point.

Normal Walking For this experiment, we conducted a straight-line test between two designated points (A and B). I personally operated the wheelchair, making a conscious effort to maintain the most direct path possible between these points. The resulting sensor data is shown in the following graph :

In Figure 5.1, the violet square highlights the initial spike created by tapping the wheel - our reference starting point for data analysis. Looking at the data after this marker, we can observe small, periodic fluctuations in the acceleration magnitude. These minor variations correspond to the regular pushing motions used to propel the wheelchair forward. The low amplitude of these fluctuations indicates that they represent typical acceleration changes associated with normal wheelchair operation, rather than unusual movements or disturbances.

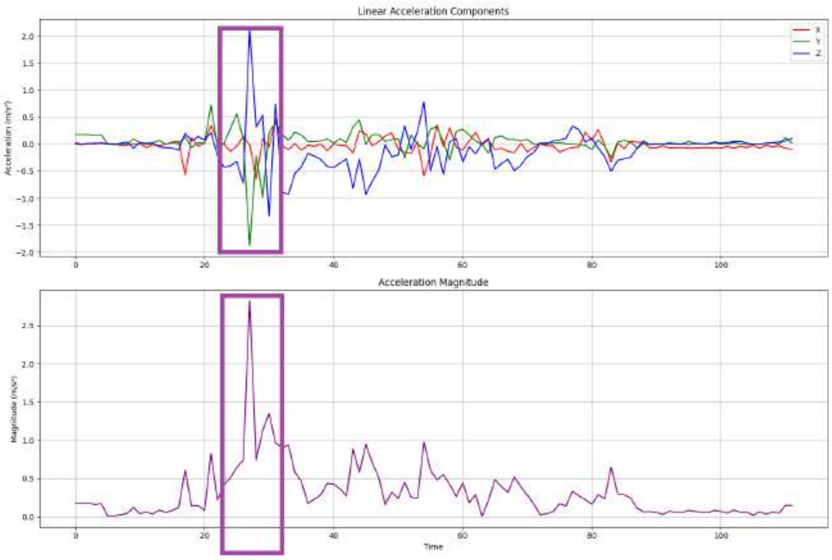


Figure 5.1: Linear Acceleration Components and Acceleration Magnitude Of Normal Use Of Wheelchair

Walking with obstacles For this test, we positioned two iron batons approximately 16 cm apart to create obstacles. The data may show some irregular spikes due to two main factors:

1. The obstacles caused some wheelchair instability.

2. The lack of prior experience with wheelchair usage.

In Figure 5.2, the yellow square highlights the moments when we encountered the obstacles with the wheelchair. Comparing this to previous data of normal wheelchair operation without obstacles, we can observe significantly higher peaks in the acceleration magnitude during these interactions. This increased magnitude clearly demonstrates the difference between the routine movement of the wheelchair and the more intense accelerations experienced when navigating obstacles.

Backward falls According to the video recording of this simulation, we noticed that unlike our standard protocol, there was no initial wheel tap recorded in the video (likely an oversight during the experiment).

The simulation went as follows :

1. Move backward until the mat

2. Fall backward on purpose

3. return carefully the chair to normal state

To provide a more comprehensive view of the wheelchair's motion, we have expanded our visualization to include a third subplot displaying the orientation components alongside the existing acceleration data (See Figure 5.3)

During the critical period between t = 45-55s, we observed significant changes across multiple sensor readings. The linear acceleration components showed dramatic activity, with X and Z components exhibiting pronounced spikes reaching approximately 8-10 m/s^2, while the Y component

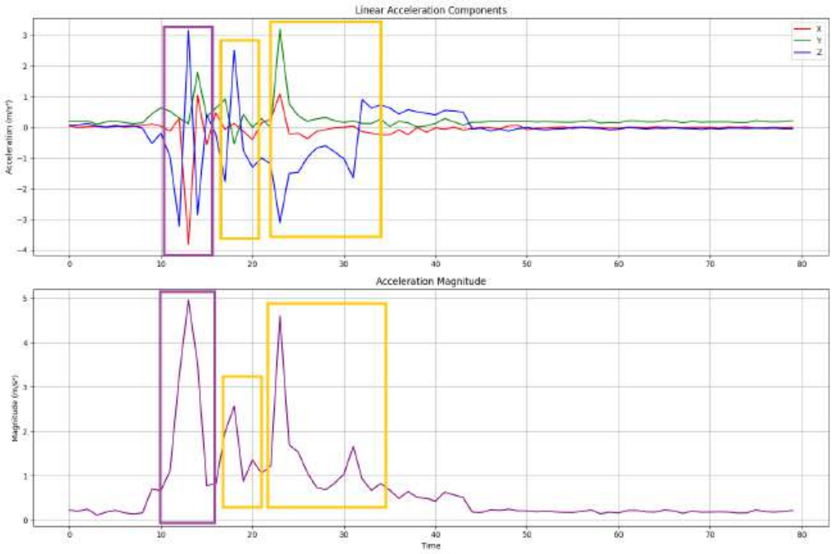


Figure 5.2: Linear Acceleration Components and Acceleration Magnitude Of Use Of Wheelchair Using Obstacles

oscillated considerably between -3 and +3 m/s^2. The acceleration magnitude reached a notable peak of approximately 13 m/s^2, marking an abrupt increase from the baseline of near 0 m/s^2. This increase in magnitude clearly indicates a sudden and violent movement of the wheelchair. Concurrent with these acceleration changes, we observed distinct variations in the wheelchair's orientation. The yaw angle underwent a significant reduction from 100° to approximately 0°, while the roll angle maintained an elevated position at approximately 350°. Notably, the pitch angle remained relatively stable near 0°. This pattern of orientation changes, occurring almost simultaneously with the acceleration spike, indicates a complex rotational movement during the fall event. Following the major event, the system entered a stabilization phase. All acceleration components gradually returned to their baseline values near 0 m/s^2, with the overall acceleration magnitude showing clear stabilization. Similarly, the orientation components settled into steady values, indicating the wheelchair had reached a stable position after the fall event.

Forward falls Near t = 60s, we observe a significant spike in acceleration components characteristic of a forward fall event. The Y component (green) exhibited the most dramatic change, reaching approximately 4 m/s^2, while the Z component (blue) showed a notable decrease to -2 m/s^2. The X component (red) showed moderate variations during this impact. These combined accelerations resulted in a substantial peak in the acceleration magnitude of approximately 4.5 m/s^2, clearly indicating the impact of the forward fall. The orientation data during this impact shows distinct changes across all components. At the moment of impact, we observe a sharp drop in the Roll (red) from its elevated position of 350°, while simultaneously the Yaw (blue) rapidly shifts from -80° to about 70°. The Pitch (green) shows minor variations but maintains a relatively stable position near 0° throughout the impact. This combination of sudden orientation changes and high acceleration magnitude is consistent with the dynamics of a forward fall impact. (See Figure 5.4)

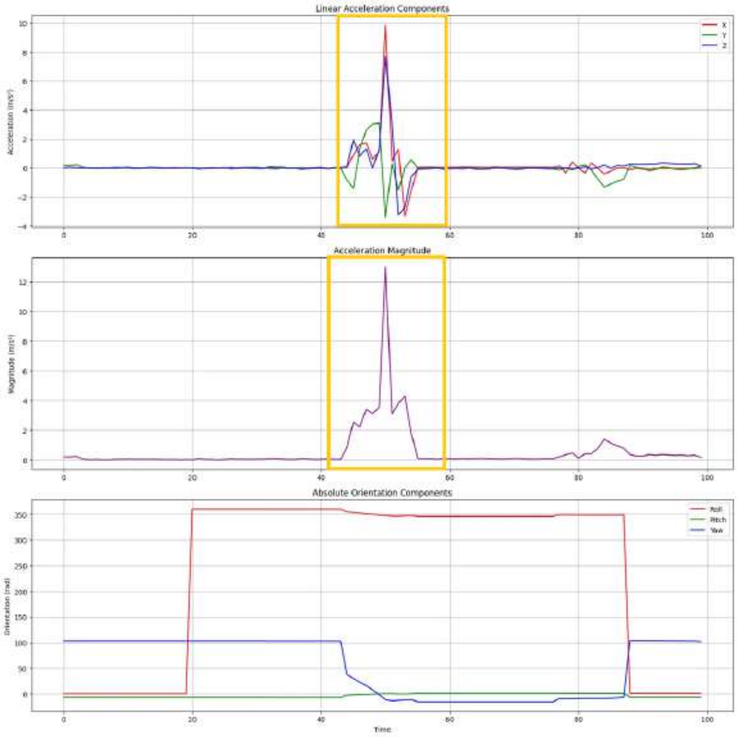


Figure 5.3: Linear Acceleration Components, Acceleration Magnitude and Orientation Component During Backward Fall Simulation

Sideways falls Near t=45-50, we observe the initial side fall impact with distinct patterns across all sensors. The acceleration data shows significant activity: the Z component (blue) exhibits a sharp drop to approximately -4 m/s², followed by multiple oscillations. The Y component (green) shows a notable spike reaching around 4 m/s², while the X component (red) displays moderate fluctuations. These combined movements result in an acceleration magnitude peak of about 6 m/s². The orientation components during this side fall are particularly telling. At the moment of impact, we see dramatic changes: the Roll (red) rapidly increases to about 220°, indicating the sideways tipping motion of the wheelchair. The Yaw (blue) simultaneously drops from its initial position of about 100° to approximately -100°, showing the rotational movement during the fall. The Pitch (green) shows slight variations but remains relatively close to 0°, which is consistent with a side fall rather than a forward or backward fall. Around t=80, we observe another set of significant spikes across all components, likely corresponding to the wheelchair being returned to its upright position. This is evidenced by the acceleration magnitude reaching approximately 8 m/s² and the Roll angle returning to its initial position, along with the Yaw stabilizing back to around 100°. (See Figure 5.5)

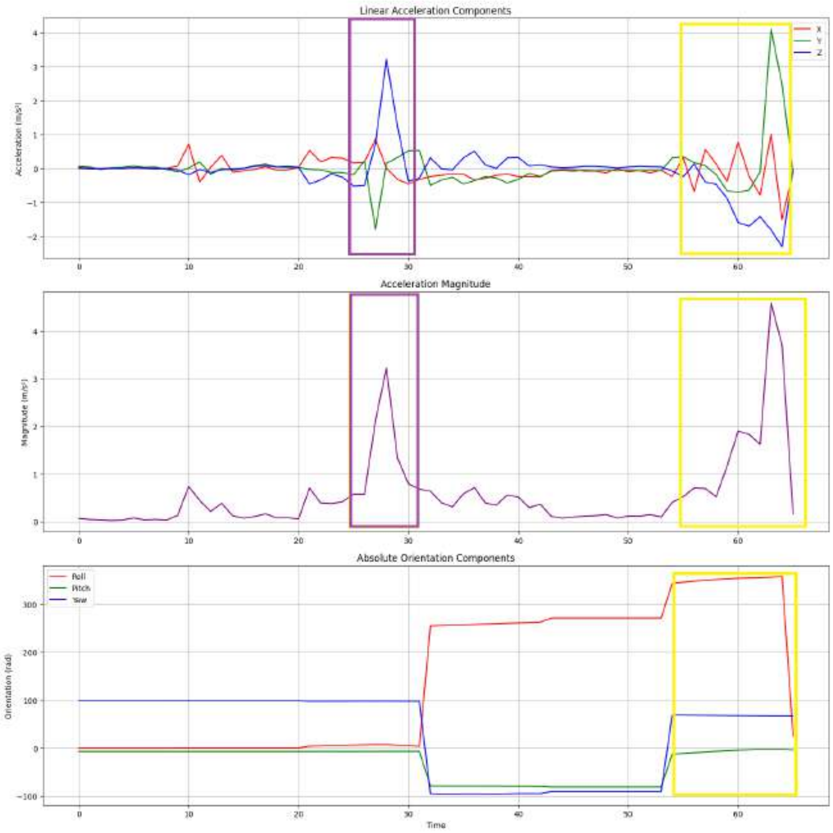


Figure 5.4: Linear Acceleration Components, Acceleration Magnitude and Orientation Component During Forward Fall Simulation

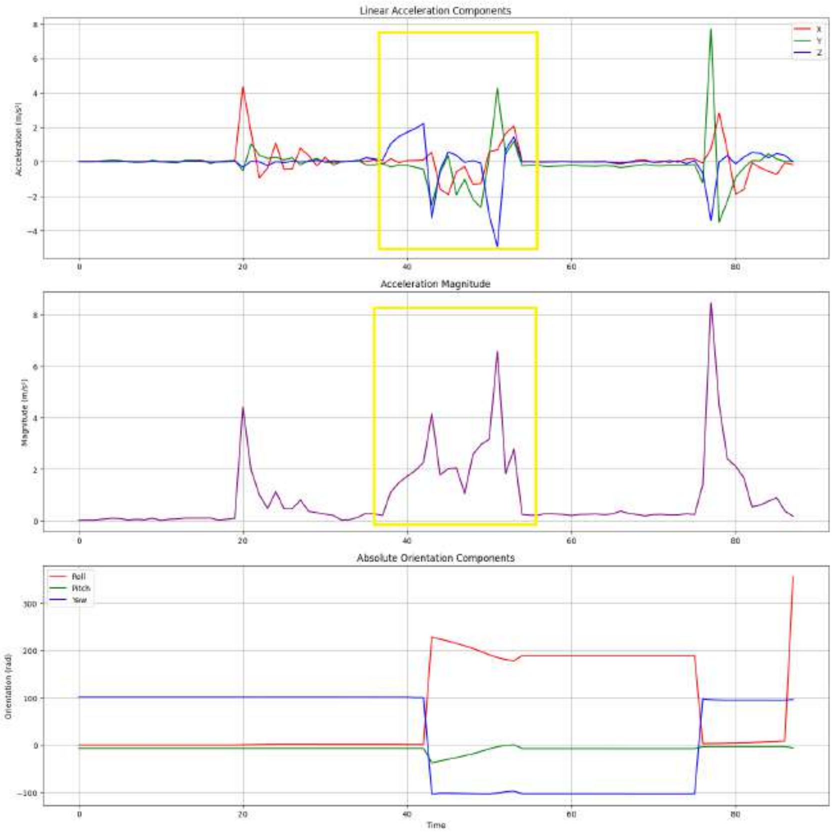


Figure 5.5: Linear Acceleration Components, Acceleration Magnitude and Orientation Component During Side Fall Simulation

5.2 Threshold Algorithm Vs OCSVM

Our results analysis will consist of conducting a comparative analysis using the same test session for each fall scenario, where both methods (One-Class SVM and threshold-based approach) will be applied to the same data. This direct comparison will allow us to interpret and contrast the effectiveness of each method under identical conditions.

We will focus on three key aspects:

1. Comparison of fall detection consistency between both methods.
2. Analysis of the methods' response to post-fall wheelchair orientation.
3. Validation of detection accuracy using video recordings.

5.2.1 Backward Fall

For our analysis, we selected a random backward fall session. As shown in Figure 5.6a, the fall event occurred approximately between timestamps 45 and 55. This interval was determined through both video analysis and sensor data, particularly noting the acceleration magnitude which peaked at 13 m/s^2. We subsequently labeled all data points within this timeframe as fall instances. It is worth noting that if fall detections were to occur slightly outside our defined boundaries, this would not necessarily invalidate the OCSVM's performance, but rather suggest that our initial manual range definition might have been too restrictive.

OCSVM Prediction

The experimental results demonstrate encouraging performance in detecting falls and abnormal positions. The plot indicates that the model accurately identified the primary fall event, which produced a peak acceleration of 13 m/s^2 at time step 45. Moreover, the model's response continued during the post-fall phase, as the inverted position of the wheelchair caused gyroscope readings to remain outside normal limits. Consequently, the model flagged these readings as abnormal position, despite the dynamic fall phase having ended. This indicates that the model not only detects the occurrence of a fall but also recognizes prolonged abnormal states of the wheelchair. The only notable concern in this case is that the model incorrectly classified an event occurring well before the fall as abnormal, even though both the video footage and sensor readings indicated normal wheelchair usage.

Threshold Algorithm Detection

The Figure 5.6c shows how our threshold-based method detects falls. The red shaded area shows when the algorithm detected a fall, with red dots marking each detection point. The method effectively catches the fall while it is happening, but stops detecting once the fall motion stops, even though it is still in an abnormal position (upside down). This is different from the OCSVM approach, as the threshold method reacts only to the actual movement of the fall and not the abnormal position afterward.

Conclusion : The threshold-based algorithm accurately detected only the fall events, whereas the OCSVM also identified events occurring after the fall, corresponding to the wheelchair remaining in an upside-down position.

5.2.2 Front Fall

Figure 5.7a illustrates the sensor data during a forward fall event. The initial impact is characterized by a sharp acceleration peak of 14 m/s^2 at time step 15, the fall event occurs within the time interval between steps 10 and 20, indicating the forward tipping motion. The acceleration magnitude plot clearly captures this impact moment. The orientation data reveals a particularly telling pattern: multiple rapid Roll angle transitions during the fall phase.

OCSVM Prediction

While we defined the fall period to begin at timestamp 10, the OCSVM's initial fall detection coincided precisely with the peak in acceleration magnitude, corresponding to the moment when the wheelchair overturned, as shown by the orientation component data. This demonstrates the model's accuracy in identifying the critical moment of the fall event. Similarly to the backward fall scenario, the model exhibited persistent fall detections after the actual event concluded, triggered by the wheelchair's sustained inverted position. However, unlike the backward fall case, it does not classify normal wheelchair usage as an anomaly.

Interpretation : This difference can be interpreted by the fact that our training dataset primarily consists of forward movement instances. Consequently, during the backward fall simulation, the backward motion—uncommon in the training data—was likely interpreted as an anomaly by the model.

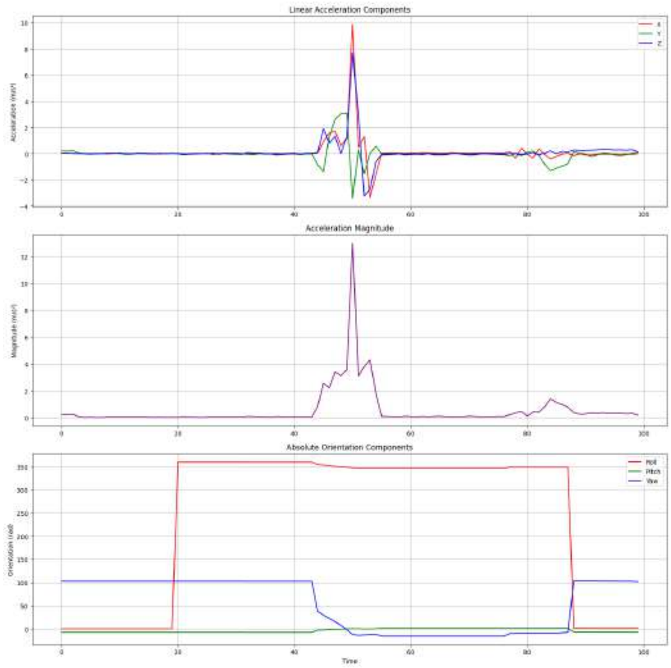
Threshold Algorithm Detection

The threshold-based approach exhibits better temporal accuracy compared to the SVM method by confining its detections strictly to the dynamic fall phase. After the initial fall event, despite minor acceleration spikes occurring at time steps 30 and 50, the system maintains its stability and does not trigger any false detections. Most importantly, this method overcomes the limitation observed in the SVM approach by correctly identifying the fall's end point, even when the wheelchair remains in an inverted position.

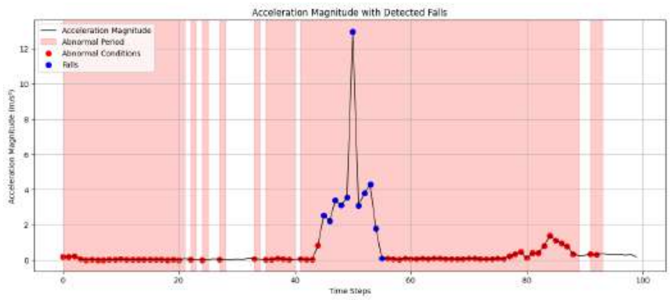
Conclusion : The OCSVM model behaved better in detecting abnormal positioning in forward falls compared to backward falls, whereas the threshold-based algorithm exhibited consistent behavior across both types of falls.

5.2.3 Normal Session

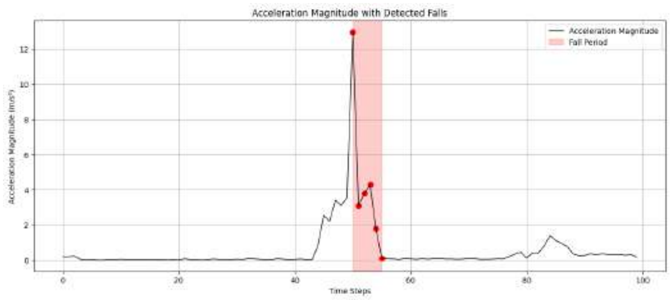
Having analyzed both methods' performance during fall events, examining their behavior during normal wheelchair operation provides additional insights into their reliability. During a session without any falls, the threshold-based algorithm demonstrated perfect specificity, generating no false detections throughout the entire sequence. In contrast, the SVM approach produced a single false positive at the beginning of the session, though it maintained accurate classifications for the remainder of the period. This comparison further highlights the relative robustness of both the threshold-based method and the OCSVM in distinguishing normal wheelchair activities from fall events (See Figure 5.8).



(a) Time series visualization of a wheelchair back fall event

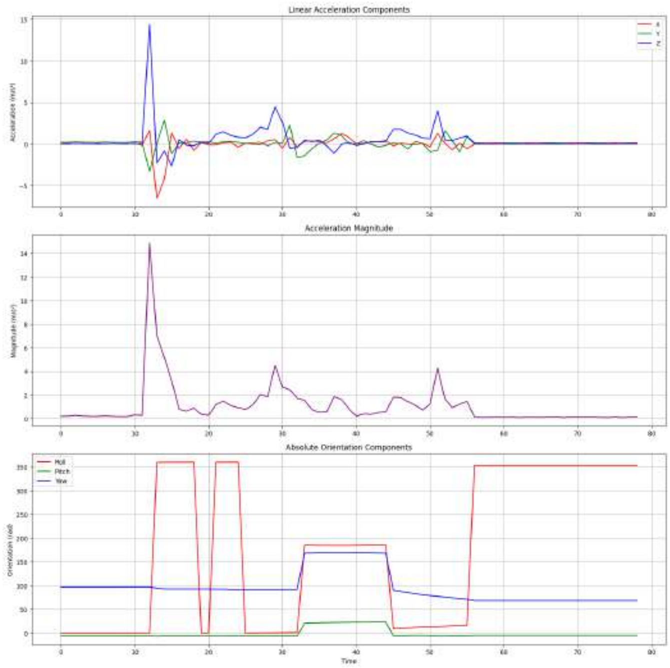


(b) OCSVM Predictions predictions for a backward fall event

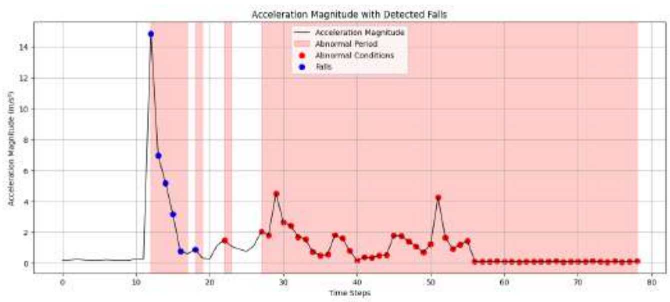


(c) Threshold Algorithm predictions for a backward fall event

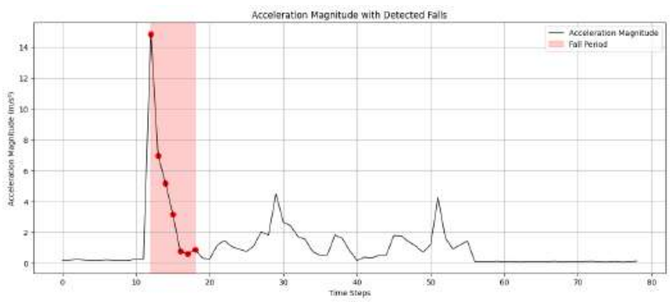
Figure 5.6: Time series visualizations of a wheelchair backward fall event, including predictions from the One-Class SVM model and detected falls using the threshold-based algorithm.



(a) Time series visualization of a wheelchair front fall event

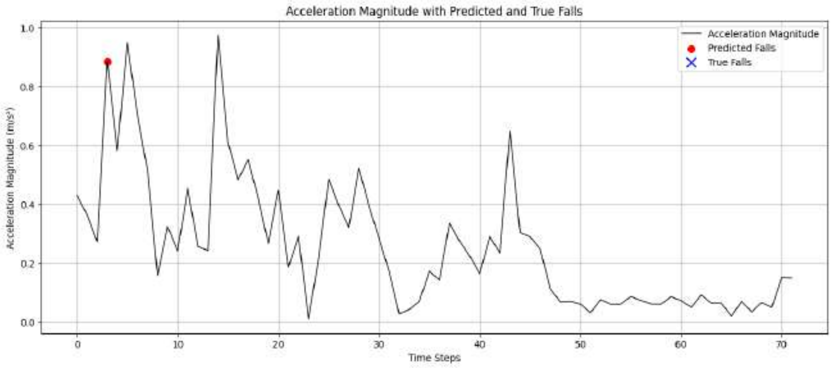


(b) One-Class SVM model predictions for a front fall event

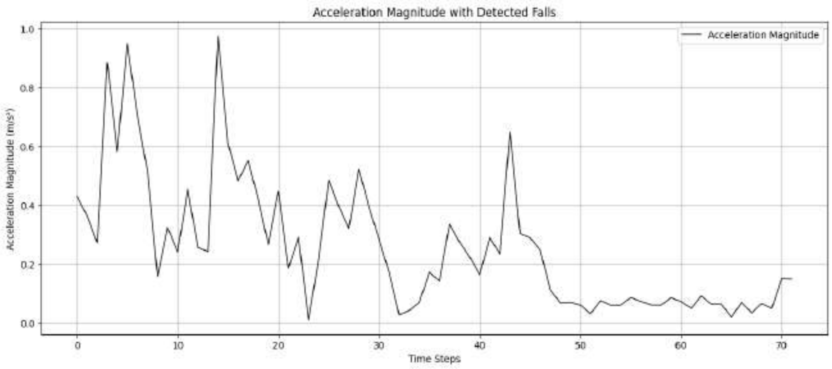


(c) Threshold Algorithm predictions for a front fall event

Figure 5.7: Time series visualizations of a wheelchair front fall event, including predictions from the One-Class SVM model and detected falls using the threshold-based algorithm.



(a) Acceleration magnitude over time with detected falls by the SVM model in a session where no fall event was present.



(b) Acceleration magnitude over time with detected falls using the threshold-based algorithm in a session where no fall event was present.

Figure 5.8: Comparison of SVM model and threshold-based algorithm predictions in a session without any fall events.

Chapter 6

Conclusions & Discussions

To conclude this thesis, the final chapter will summarize the work completed, highlighting the contributions made to this large project and discussing possible future improvements and developments.

As our work builds on previous efforts, we started by reviewing the earlier work done, which involved selecting and configuring sensors to collect acceleration data and other temperatural informations. This data is fundamental, serving as the foundation for our entire thesis, which focuses on detecting falls and abnormal wheelchair positions.

We started by designing and developing the application with various functionalities tailored to wheelchair users to simplify their daily lives. These features include route planning with accessible paths, gamifying achievements to encourage user engagement, and enabling Bluetooth Low Energy (BLE) connectivity between the mobile device and the sensor box.

The core focus of our work was on detecting falls and abnormal wheelchair positions. The most challenging aspect was data collection, as we needed to devise a reliable method to securely attach the sensors to the wheelchair. This ensured that (1) the collected data was usable for analysis, and (2) it could effectively simulate how the final application would respond in real-world scenarios. After collecting the data, we implemented two methods for fall and abnormal position detection. The first method was a threshold-based algorithm, where we continuously checked whether the sensor readings exceeded a predefined threshold and made decisions based on these exceedances. The second method involved implementing a One-Class Support Vector Machine (OCSVM) for anomaly detection. The results were promising: the threshold-based algorithm accurately detected all fall events but stopped detecting once the wheelchair's motion ceased, meaning it couldn't identify abnormal positioning. On the other hand, the OCSVM not only detected the falls but also recognized when the wheelchair remained upside down. However, it also showed some false positive predictions.

Future Work With the fall detection method implemented and a solid foundation for the application established, future contributions could focus on the following:

1. Enhancing the fall detection system by collecting additional data in real-world outdoor environments rather than controlled laboratory settings. This would allow for more accurate and realistic setups to train the models.

2. Extending the sensor-to-device connection to enable seamless retrieval of sensor data in real-time.

3. Developing the routing feature with the integrated OpenStreetMap API and utilizing GPS sensor readings to provide accurate and practical navigation for wheelchair users.

Bibliography

Doukhan Abdulaziz. Collecting informative, high-quality, and time-consistent data on a manual wheelchair with a cost-effective microcontroller system. Master's thesis, Vrije Universiteit Brussel, June 2024.

Mathieu ABET. Quelle est la différence entre bluetooth et bluetooth low energy ?, 2021. URL `https://elainnovation.com/difference-entre-bluetooth-et-bluetooth-low-energy/`. Accessed: [01-12-2024].

Mars Ballantyne, Archit Jha, Anna Jacobsen, J. Scott Hawker, and Yasmine N. El-Glaly. Study of accessibility guidelines of mobile applications. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*, MUM '18, page 305–315, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450365949. doi: 10.1145/3282894.3282921. URL `https://doi.org/10.1145/3282894.3282921`.

J. Bennett. *OpenStreetMap*. Packt Pub., 2010. ISBN 978-1-84719-751-1. URL `https://books.google.be/books?id=SZf...`

A.K. Bourke, J.V. O'Brien, and G.M. Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Posture*, 26(2):194–199, 2007. ISSN 0966-6362. doi: https://doi.org/10.1016/j.gaitpost.2006.09.012. URL `https://www.sciencedirect.com/science/article/pii/S0966636206001895`.

John Clarkson and Roger Coleman. Inclusive design. *Journal of Engineering Design*, 21 (2-3):127–129, 2010. doi: 10.1080/09544821003693689. URL `https://doi.org/10.1080/09544821003693689`.

Rory A. Cooper, Rosemarie Cooper, and Michael L. Boninger. Trends and issues in wheelchair technologies. *Assistive Technology*, 20(2):61–72, 2008. doi: 10.1080/10400435.2008.10131933. URL `https://doi.org/10.1080/10400435.2008.10131933`. PMID: 18646429.

Homa Foroughi, Alireza Rezvanian, and Amirhossien Paziraee. Robust fall detection using human shape and multi-class support vector machine. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pages 413–420, 2008. doi: 10.1109/ICVGIP.2008.49.

Adrian Furnham and Rebecca Thompson. Actual and perceived attitudes of wheelchair users. *Counselling Psychology Quarterly*, 7(1):35–51, 1994. doi: 10.1080/09515079408254133. URL `https://doi.org/10.1080/09515079408254133`.

GeeksforGeeks. Support vector machine algorithm. `https://www.geeksforgeeks.org/support-vector-machine-algorithm/`. Accessed: [your date of access].

Robin Heydon and Nick Hunn. Bluetooth low energy. *CSR Presentation, Bluetooth SIG https://www. bluetooth. org/DocMan/handlers/DownloadDoc. ashx*, 2012.

Interaction Design Foundation - IxDF. What is user centered design (ucd)? https://www.interaction-design.org/literature/topics/user-centered-design, June 5 2016.

Peter Presti Jacob J. Sosnoff Harshal P. Mahajan Mikaela L. Frechette Libak Abou, Alexander Fliflet and Laura A. Rice. Fall detection from a manual wheelchair: preliminary findings based on accelerometers using machine learning techniques. *Assistive Technology*, 35(6):523–531, 2023. doi: 10.1080/10400435.2023.2177775. URL https://doi.org/10.1080/10400435.2023.2177775. PMID: 36749900.

Dongha Lim, Chulho Park, Nam Ho Kim, Sang-Hoon Kim, and Yun Seop Yu. Fall-detection algorithm using 3-axis acceleration: Combination with simple threshold and hidden markov model. *International Journal of Distributed Sensor Networks*, 2014:1–8, 2014. doi: 10.1155/2014/896030.

Heeket Mehta, Pratik Kanani, and Priya Lande. Google maps. *International Journal of Computer Applications*, 178(8):41–46, 2019.

Gilani Mobility. How many wheelchair users are there in the world?, n.d. URL https://www.gilanimobility.ae/how-many-wheelchair-users-in-the-world/?srsltid=AfmBOoqSugaaYuUGBh0MrcsX58d7cyXbWoAKJg2OYhswB5Rn3Sy6xAK4. Accessed: 2024-11-29.

Louise et al Moody. Improving wheelchair prescription: An analysis of user needs and existing tools. *Proceedings of [Add conference/journal name if known]*, pages 1980–1984, January 2012. 10.3233/WOR-2012-0418-1980.

Ignas Plauska, Agnius Liutkevičius, and Audronė Janavičiūtė. Performance evaluation of c/c++, micropython, rust and tinygo programming languages on esp32 microcontroller. *Electronics*, 12(1):143, 2022.

Md. Atiqur Rahman, S.M. Ahsanuzzaman, Abid Hasan, Ishman Rahman, Toufik Ahmed, and Md Monirul Kadir. Building a wheelchair controlling and fall detection system using mobile application. In *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, pages 213–218, 2020. doi: 10.1109/ICAICT51780.2020.9333478.

D. Razum, G. Seketa, J. Vugrin, and I. Lackovic. Optimal threshold selection for threshold-based fall detection algorithms with multiple features. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1513–1516, 2018. doi: 10.23919/MIPRO.2018.8400272.

Toni Saia, Emily Vogel, and Sadie Salazar. "we need a world we can operate in": Exploring the relationship between societal stigma and depression among wheelchair users. *Disability and Health Journal*, 17(3):101624, 2024. ISSN 1936-6574. doi: https://doi.org/10.1016/j.dhjo.2024.101624. URL https://www.sciencedirect.com/science/article/pii/S1936657424000554.

D. A. Sanders, M. Langner, and G. E. Tewkesbury. Improving wheelchair-driving using a sensor system to control wheelchair-veer and variable-switches as an alternative to digital-switches or joysticks. *Industrial Robot*, 37(2):157–167, 2010. doi: 10.1108/01439911011018939. URL https://doi.org/10.1108/01439911011018939.

Klaus Schilling, Hubert Roth, Robert Lieb, and Hubert Stützle. Sensors to improve the safety for wheelchair users. In *Proceedings of the 3rd Annual TIDE-Technology for Inclusive Design and Equality-Congress*, pages 327–331, Helsinki, Finland, 1998.

S.Y. Sheikh and M.T. Jilani. A ubiquitous wheelchair fall detection system using low-cost embedded inertial sensors and unsupervised one-class svm. *Journal of Ambient Intelligence and Humanized Computing*, 14:147–162, 2023. doi: 10.1007/s12652-021-03279-6.

Maria Shitkova, Justus Holler, Tobias Heide, Nico Clever, and Jörg Becker. Towards usability guidelines for mobile websites and applications. In *Wirtschaftsinformatik Proceedings 2015*, page 107, 2015. URL `https://aisel.aisnet.org/wi2015/107`.

D. A. Sterling, J. A. O'Connor, and J. Bonadies. Geriatric falls: injury severity is high and disproportionate to mechanism. *The Journal of Trauma*, 50(1):116–119, 2001. doi: 10.1097/00005373-200101000-00021.

Haifeng Se Tao Xu and Jiahui Liu. A fusion fall detection algorithm combining threshold-based method and convolutional neural network. *Microprocessors and Microsystems*, 82:103828, 2021. ISSN 0141-9331. doi: https://doi.org/10.1016/j.micpro.2021.103828. URL `https://www.sciencedirect.com/science/article/pii/S0141933121000089`.

Sara Usmani, Abdul Saboor, Muhammad Haris, Muneeb A. Khan, and Heemin Park. Latest research trends in fall detection and prevention using machine learning: A systematic review. *Sensors*, 21(15), 2021. ISSN 1424-8220. doi: 10.3390/s21155134. URL `https://www.mdpi.com/1424-8220/21/15/5134`.

World Health Organization. *Global report on health equity for persons with disabilities*. World Health Organization, 2023. URL `https://www.who.int/news-room/fact-sheets/detail/disability-and-health`.

Miao Yu, Yuanzhang Yu, Adel Rhuma, Syed Mohsen Raza Naqvi, Liang Wang, and Jonathon A. Chambers. An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. *IEEE Journal of Biomedical and Health Informatics*, 17(6):1002–1014, 2013. doi: 10.1109/JBHI.2013.2274479.