Alma Mater Studiorum · Università di Bologna

Department of
Computer Science And Engeneering

Second Cycle Degree in Artificial Intelligence

MACHINE LEARNING AND DEEP LEARNING FOR INTELLIGENT INSOLES

master thesis in
deep learning

Supervisor:
Prof.
ANDREA ASPERTI

Co-Supervisor:
Dott. Ing.
NIDHAL LOUHICHI

Candidate:
PRIMIANO ARMINIO CRISTINO

II Session - I Appeal

Academic Year 2022/2023

This page is intentionally left blank.

To my dear parents

your support and teachings have been the lifeblood
that has shaped the person I am today.
Without you, all of this would have remained a dream.
I am eternally grateful to you.

Contents

Abstract

The field of pervasive healthcare relies heavily on mobile personal sensing technology to identify everyday human activities. One company, eSteps Inc., headquartered in Bologna, Italy, with American origins, is actively working to address the increasing motor disabilities affecting the lower limbs. They provide comprehensive monitoring solutions that cover pre-hospitalization, hospitalization, and post-hospitalization phases, all based on biomechanics and telerehabilitation protocols. This paper's main goal is to develop an Artificial Intelligence (AI) model. The AI model's purpose is to accurately recognize the specific activities performed by individuals, whether they have Multiple Sclerosis or are in good health. The model uses data collected from eSteps' innovative devices and aims to significantly enhance the quality of care and support for patients with motor disabilities.

Chapter 1

Introduction

Gait analysis plays a pivotal role in monitoring various diseases, with Multiple Sclerosis (MS) being a prominent example. While the Expanded Disability Scale Score (EDSS) has traditionally been the primary measure of disability in MS studies, it has limitations in assessing motor disability. Fortunately, recent technological advancements have enabled the collection of data using wearable inertial sensors. These sensors, due to their small size, portability, and affordability, can be used not only in laboratory settings but also for real-world monitoring, offering valuable insights into patients' daily lives.

By analyzing data from Inertial Measurement Units (IMUs), researchers have identified significant differences in gait parameters between MS patients and healthy individuals. Moreover, these differences often correlate with increasing EDSS levels, highlighting the critical role of gait analysis in tracking disease progression and selecting appropriate therapies. The importance of continuous monitoring of daily activities, supported by effective algorithms, becomes really important. In fact fewer daily steps are associated with higher EDSS-measured disability.

Artificial Intelligence (AI) plays a crucial role in the continuous monitoring of diseases, With the help of AI, medical practitioners can effectively track the progression of illnesses, allowing for earlier detection and intervention. AI-driven system can detect patterns and anomalies that might elude traditional diagnostic methods. This way, the incorporation of AI into

disease monitoring, can lead, not only to the development of personalized treatment plans, but to an effective healthcare system.

1.1 Problem

Deducing human actions from sensor data is a complex challenge. Designing a robust activity recognition system involves various important factors. Each of these elements plays a crucial role in creating a reliable system for recognizing human activities. One of the most critical factors in Human Activity Recognition (HAR) is the quality and relevance of the features used to represent the data.

Imagine HAR as a puzzle where the goal is to identify and classify various human activities accurately. The pieces of this puzzle are the data collected from sensors, capturing information about movements, positions, and environmental conditions. To assemble a complete picture, we need to carefully select and craft these pieces – that's where feature quality and relevance come into play. Features are numerical representations of specific aspects of the data, such as speed, orientation, or frequency patterns. Not all features are equally useful for all types of activities and some of them could be noisy, worsening the quality of these.

The second critical aspect is the choice the right model because it determines how effectively the system can make sense of the data. Moreover, the model's architecture, whether it's based on Machine Learning or Deep Learning, impacts greatly the system's performance.

In this paper we'll explore the core components of this fascinating discipline, from data acquisition and preprocessing to feature extraction and model selection.

1.2 eSteps

eSteps Inc. has developed a sophisticated, customized device for monitoring motor activity, assessing fatigue, and identifying injury risks in patients. They've also created a specialized mobile app that allows data sharing

among patients, caregivers, and medical specialists.

> Our mission is to enhance the quality of life, manage symptoms, and elevate overall well-being for individuals with neurodegenerative diseases, through innovative technology and personalized care. We empower individuals to reclaim control over their well-being and embark on a journey of progress.

1.3 Structure

- **Chapter 1:** This chapter offers a brief overview of the thesis topic, what problem it aims to address, and the solution it proposes.
- **Chapter 2:** This chapter goes into the details of Multiple Sclerosis (MS) as a disease and provide a review of the current research on Human Activity Recognition (HAR).
- **Chapter 3:** This chapter explains the Machine Learning and Deep Learning models used and the results obtained.
- **Chapter 4:** This chapter summarizes the most important discoveries from the research and suggests areas for future studies.

Chapter 2

State of the art

2.1 Multiple Sclerosis

Multiple Sclerosis (MS) is a chronic autoimmune neurological and neurodegenerative disorder that affects the Central Nervous System (CNS) [1]. It is characterized by an exaggerated immune response that mistakenly targets components of the CNS as foreign invaders. This immune reaction results in the damage of both axons, the nerve fibers transmitting electrical impulses between neurons, and myelin, the protective sheath surrounding axons. Oligodendrocytes, specialized cells responsible for myelin production, are also attacked by the immune system. This process, known as demyelination, can lead to chronic CNS inflammation. Demyelination can manifest initially as plaques during the early stages of MS, where axons are demyelinated, followed by periods of remyelination.

Repeated cycles of demyelination and remyelination can result in the formation of scars in the white matter of the CNS leading to chronic damage, which characterizes the sclerosis stage of the disease. The course of MS can vary widely among individuals, and the disease is known for its unpredictability [2].

In addition to the autoimmune response, which is the primary driver of MS, there are various factors that can potentially trigger demyelination in susceptible individuals, such as genetic predisposition, toxins, hormonal

factors, smoking and viral infections.

The degree of myelin and axonal destruction can vary, leading to different categories or phases of MS:

- **Relapsing-Remitting**: This is the most common phase, characterized by periods of disease progression (demyelination) followed by periods of remission (remyelination).
- **Primarily Progressive**: In this form of MS, symptoms steadily worsen over time without periods of remission, occasionally plateaus can occur.
- **Secondarily Progressive**: In some cases, individuals with relapsing-remitting MS may experience a progression of the disease with or without periods of remission. This phase may involve stabilization of symptoms during plateaus (period in the disease's progression where the symptoms remain relatively stable without significant improvement or worsening).
- **Progressively-Relapsing**: This is a rare form of MS in which the disease continuously worsens from the beginning, with intermittent flare-ups of symptoms. There are no periods of remission or plateaus.

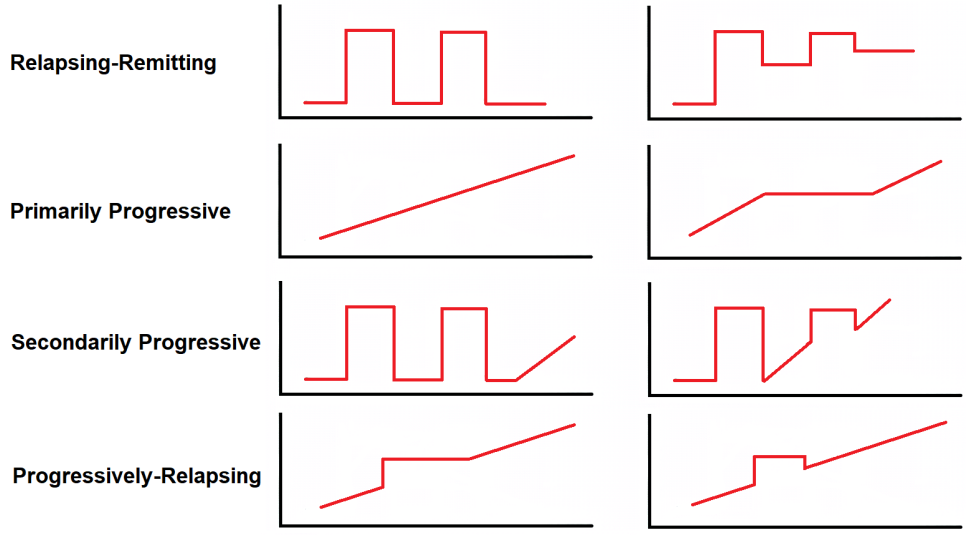


Figure 2.1: Multiple sclerosis phases

It's important to understand these categories and the factors that contribute to MS because it helps the creation of treatments and interventions to meet the specific needs of individuals affected by MS. Detect MS is in fact not an easy task. It typically involves a combination of clinical evaluation, medical tests, and diagnostic criteria. Symptoms [3] of MS can vary among patients, even within the same MS category. While numerous symptoms can manifest, some of the most common ones include:

- **Fatigue**: Overwhelming tiredness and lack of energy are common in MS.
- **Numbness or Tingling**: Patients may experience a sensation of pins and needles, numbness, or tingling in various parts of the body.
- **Muscle Weakness**: Weakness in the limbs, making it difficult to move or control muscles, is a typical symptom.
- **Balance Problems**: MS can affect balance and coordination, leading to difficulty walking or maintaining stability.

- **Muscle Spasms and Stiffness**: Involuntary muscle contractions and stiffness are common.
- **Vision Problems**: Vision-related symptoms include blurred or double vision, pain in the eyes, and difficulty focusing.
- **Cognitive Changes**: Memory problems, difficulty concentrating, and alterations in cognitive processes.
- **Bladder and Bowel Issues**: MS can result in urinary urgency, frequent urination, incontinence, and constipation.
- **Emotional Changes**: Mood changes, including depression, mood swings, and heightened emotional sensitivity.
- **Speech and Swallowing Difficulties**: Speech difficulties and problems with swallowing can also manifest.

Ongoing research studies continue to explore these potential triggers and their roles in the development of the disease as well as the distinct phases of MS. It's important to note that not all individuals with these risk factors will develop MS.

2.2 Human Activity Recognition

2.2.1 Introduction

Human Activity Recognition (HAR) is the task of gathering, examining, and categorizing human actions. It is a challenging research study with significant interest for several reasons. First of all, it's a significant field of research because it has the potential to improve our interactions with computers and plays a crucial role in monitoring our health., especially for issues like motor diseases, as previously discussed.

HAR encounters specific challenges [4] that must be overcome to construct high-performance systems. Initially, there are general challenges common to most pattern recognition tasks, including variations within the

same activity (intraclass variability), similarities between different activities (interclass similarity), and the issue of identifying irrelevant activities (null class problem).

Furthermore, HAR faces unique challenges:

- Clearly defining physical activities.
- Addressing class imbalances, where some activities occur more frequently than others, requires careful handling of data.
- Ensuring accurate ground truth annotation.
- Thoughtful dataset design is essential to create effective and representative training and evaluation datasets.

2.2.2 Challenges

2.2.2.1 Intraclass variability

This situation, known as "intraclass variability," occurs when the same activity can be executed differently, not only across various individuals (inter-person variability) but also by the same individual on different occasions (intra-person variability). Various factors, including the environment, emotions, stress, and fatigue, contribute to these variations.

When HAR systems are exclusively trained on data from a single subject (referred to as subject-dependent systems), they can enhance their ability to handle intra-person variability simply by increasing the amount of training data. Conversely, for HAR systems trained on data from multiple subjects (referred to as subject-independent systems), which need to address inter-person variability, it's advisable to incorporate features that are not person-specific. Regardless of the approach chosen, augmenting the volume of data is always a beneficial strategy.

2.2.2.2 Interclass similarity

Conversely, there's another challenge known as the opposite of intraclass variability. This challenge arises when distinct activities share some similar-

ities, complicating the recognition task. Overcoming these similarities is a critical aspect of the feature extraction process.

2.2.2.3 Null class problem

This challenge highlights the issue that not all data acquired are useful for tasks like HAR, where patterns need to be recognized. Among the data, there are often activities that are irrelevant and can be easily misclassified,

These irrelevant activities are classified as part of the "null class." Identifying and managing this null class is a challenging problem because there is currently no specific approach to avoid it. Detecting null activities can be accomplished by studying and distinguishing signal variances.

2.2.2.4 Physical activity

Establishing a precise understanding of the unique characteristics of each activity is fundamental to building more effective recognition systems. In particular, some HAR systems may prioritize the quality with which the activities are performed.

2.2.2.5 Class imbalance

When acquiring new data not all activities occur with the same frequency. Typically, certain activities are more frequent than others. Consequently, after the data collection phase, some activities may be underrepresented compared to others.

To mitigate this issue, there are several approaches available. The simplest method involves increasing the amount of data during the training process, while also considering the removal of overrepresented data to rebalance the dataset.

2.2.2.6 Ground truth label

Collecting annotated or "ground truth labeled" training data is a resource-intensive and laborious task. Annotators are required to meticulously label the data in real-time, which can be both costly and time-consuming.

2.2.2.7 Data collection

Unlike some other research domains like speech recognition or computer vision, the HAR research community has yet to initiate a collective effort to amass comprehensive and versatile datasets. Furthermore, acquire datasets is challenging. Some factors substantially affect performance like high data quality, multiple modalities or sensors, long-term recording durations, or substantial participant numbers. It must be also considered a trade-off between sensor ease of use and the time for preparing, conducting, the experiment, and the logistical considerations for participants in the experiment.

2.2.3 Vision based

Human Activity Recognition holds a crucial role in various Computer Vision domains, including tasks such as identifying humans in videos, estimating their poses, and tracking their movements. Vision-based systems in HAR process data represented in a three-dimensional space-time framework, rather than the conventional two-dimensional space.

It's important to emphasize that the features extracted from images and videos must capture any alterations in human movements. These video-based systems are tied to the type of data and consequentially to features extracted. To enhance the quality of recognition systems, it's essential to utilize only pertinent features. Feature representation and selection have long been historical challenges in Machine Learning and Computer Vision. However, there are two particularly aspects to take in considerations:

- **Interaction recognition** refers to situations where multiple action and interactions occur within the same input sequence.
- **Action detection** it's about determining when and where a particular activity occurs within a sequence of data. Precisely, involves pinpointing the precise location of specific activities within the spatiotemporal context of the input data.

2.2.3.1 Approaches

In recent years, numerous approaches have been proposed to address this challenge, including spatial-temporal features and features derived from changes in motion and posture. These challenges underscore the need for advanced techniques and models to accurately capture interactions in complex real-world scenarios.

Vision-based Human Activity Recognition systems typically favor the use of Deep Learning approaches over traditional Machine Learning or unsupervised learning methods [5]. In fact, a considerable amount of research is dedicated to improve neural networks. These networks are commonly structured as a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

The main approaches consist of:

- **Two-stream convolutional network [6]**: optical flow information is computed from the image sequence. The image and optical flow sequences are then separately fed into two convolutional neural networks (CNNs) during the model training phase. Fusion takes place in the final classification layer of the network. The two-stream network receives inputs consisting of a single-frame image and a multi-optical flow frame image stack, and the network employs 2D image convolution for processing. []

- **3D convolution network [7]**: treats the video data as a 3D space-time structure and employs a 3D convolution technique to extract and learn features related to human actions.

- **Spatiotemporal LSTM model [8] [9]**: extending recurrent neural networks into spatiotemporal domains enables the analysis of concealed sources of action-related information.

The preference for Deep Learning, particularly the combination of CNNs and RNNs, is driven by its ability to effectively capture temporal patterns and intricate spatial features in vision-based HAR systems. This approach has proven to be highly efficient and capable of delivering accurate results in classifying human activities from video and image sequences.

2.2.3.2 Drawbacks

While Vision-based Human Activity Recognition has showcased the development of robust and high-performance systems, it does come with certain drawbacks.

- vision-based HAR systems necessitate the collection of data using devices like cameras, which can be notably costly.
- the vision-based approach introduces additional complexity to the image processing pipeline.
- in intricate scenarios, such as those captured by intelligent video surveillance systems, abnormal actions are frequently linked to the objects or attachments carried by individuals. This issue also intersects with the challenge of interaction recognition.

2.2.4 Sensor based

Sensor-based Human Activity Recognition operates with data originating from sensors. These wearable sensors can capture data from subjects and are particularly suitable for capturing human movements.

Most of the sensors avaible for HAR consists in:

- **Accelerometers**
- **Gyroscopes**
- **Magnetometers**
- **Barometers**
- **Temperature sensors**

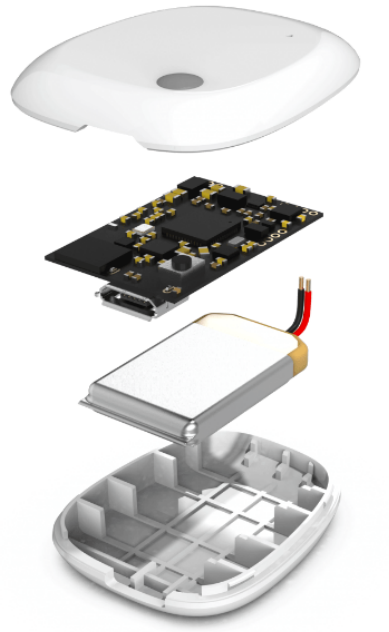


Figure 2.2: MetaMotionS components

Accelerometers and gyroscopes, and sometimes magnetometers, are often integrated into devices known as Inertial Measurement Units (IMUs), which are widely used due to their cost-effectiveness.

Nowadays, these sensors are embedded in devices such as smartphones and watches. Consequently, many HAR studies have leveraged smartphones. Today there are many apps that collect this type of data. As smartphones continue to grow as an extension of our daily lives, so will the expectation that they have greater inclusion within our workspace.

2.2.4.1 Tradeoffs

Collecting data from sensors introduces some challenges for HAR researchers. First of all, environmental conditions can affect the data acquisition process making similar feature completely different. Secondly, failures in acquisition process, being jumps in the data or complete turns off, can lead to a low

performance. Furthermore, sensors embedded to specific devices could not be able to acquire the best fluctuations in the data variance. Also, subtle deviations over time, such as signal drift, can be far more challenging. Some sensors are especially sensitive to environmental factors, like barometers requiring frequent recalibration or magnetometers being susceptible to ferromagnetic influences. Additionally, when sensors are incorporated into portable devices such as mobile phones, they may be utilized in various ways or positioned differently on the body.

For many real-world applications like gesture-based input, there's a need for real-time signal processing and classification. However, for tasks like behavioral monitoring or trend analysis over extended periods, offline data analysis and classification may suffice. Some HAR systems demand low-latency classification and immediate feedback, while others can tolerate higher latency. Embedded sensors, particularly those with limited processing power, pose an additional challenge. One solution to address these challenges is to introduce a central component in the experimental setup. This central component aggregates, processes, and fuses information from different sensors.

2.2.4.2 Methodology

There is no universal standard for the activities to be detected or how they should be characterized. Therefore, some guidelines are based on key considerations:

- **Execution**: Systems can record data for activity detection either in a post-processing manner (offline) or in real-time (online).
- **Generalization**: Systems can be designed to work with various subjects (subject-independent) or optimized for a specific subject (subject-specific). They should also exhibit robustness to temporal variations caused by external conditions.
- **Recognition**: Systems can automatically detect activities from the input sequence or segment the input sequence and generate activity labels corresponding to each segment.

- **Activity**: Systems can detect periodic activities (e.g., walking, running), sporadic activities (e.g., gestures), or static activities (e.g., standing, sitting).
- **System Model**: Systems can employ an environmental model to aid in activity detection (stateful), or they can rely solely on the raw acquired data (stateless).

HAR offers more degrees of freedom in terms of system design and implementation.

The activity recognition chain (ARC) [10] is a sequence of five sequential stages:

- **Data acquisition**
- **Preprocessing**
- **Data Segmentation**
- **Feature extraction**
- **Classification**

The ARC ensure that HAR systems are robust and accurate in recognizing various human activities.

2.2.4.3 Data acquisition

The process begins with the collection of data, which may come from sensors like accelerometers, gyroscopes, cameras, or other sources. These sensors capture information related to human movements and activities [11].

Figure 2.3: MetaMotionS and MetaBase

2.2.4.4 Preprocessing

Collected data often require preprocessing to clean, filter, and format it for analysis. This step involves removing noise, handling missing data, ensuring data consistency and data outlier detection.

Noise in signal processing refers to unwanted and uncorrelated components that introduce undesirable changes to the original signal.

Wavelet transform methods offer an effective approach to represent and process signals. It's able to extract meaningful information from noisy signals. One of the key advantages of wavelet transformation is its ability to preserve the essential coefficients of the original signal while removing noise components. This is achieved through a process called thresholding, where noise signals are identified and their coefficients are adjusted accordingly. The choice of an appropriate thresholding scheme is crucial for achieving

optimal noise reduction.

Imputation is an important preprocessing task in data analysis for dealing with incomplete data. Many machine learning algorithms assume that the data are complete. Since missing data are common, it can lead to inaccurate or unreliable results. The first step is to determine why the data is missing. Common reasons include poor network connectivity, faulty sensor systems, environmental factors, and synchronization issues. Based on the identified reasons and patterns, an algorithm to handling them is constructed.

There are several algorithms commonly used to handle missing data:

- **Mean, Median, or Mode**: In this simple technique, missing values are replaced with the mean (average), median (middle value), or mode (most frequent value) of the observed data
- **Interpolation**: Interpolation methods estimate missing values based on the values of adjacent data points.
- **Gaussian Mixture model (GMM)**: GMM is a probabilistic model used in machine learning and statistics that uses a soft clustering approach for distributing the data points to different clusters. It assumes that the probability distribution of the data is generated from a weighted sum of multiple Gaussian distributions.

  Filtering is a crucial signal processing technique in the context of Human Activity Recognition. It facilitates the extraction of meaningful features from raw sensor data. By attenuating high-frequency components in the data, low-pass filtering effectively removes noise and rapid fluctuations, allowing for a more precise analysis of the data signal, helping the motion patterns recognition.

2.2.4.5 Data segmentation

Collected data are typically continuous streams of information. Data segmentation involves breaking these continuous streams into discrete segments or windows [12]. Each segment represents a specific timeframe during which

an activity occurs. Accurate segmentation is essential to isolate individual activities for analysis.

Often, the exact boundaries of an activity are difficult to define. In the literature, several methods are available to address the challenge of data segmentation.

- **Sliding window segmentation** involves dividing the continuous data stream into overlapping or non-overlapping windows of fixed or variable size. The window slides over the data at a predefined step size. This method can capture detailed information within each window, making it suitable for recognizing short and dynamic activities.

- **Energy-based segmentation** focuses on variations in signal energy to detect activity boundaries. Changes in signal energy, often derived from sensor data, that can indicate the start or end of an activity.This can be effective when activities are characterized by distinct energy patterns.

- **Rest-position segmentation** identifies periods of inactivity or rest between activities. It assumes that during inactivity, the sensor data remains relatively constant or falls within a specific range. This method is valuable when recognizing activities with distinctive resting or idle periods. For instance, it can be useful in monitoring activities like sitting, standing, and walking, where the transitions between rest and motion are significant.

2.2.4.6 Feature Extraction

Feature extraction involves identifying relevant patterns or features within the data that can be used for activity recognition. These features may include temporal, spatial, or frequency-domain characteristics of the data.

One approach is to automatically calculate features directly from the sensor data. These features are typically derived using mathematical and signal processing techniques. For example, features like mean, variance, skewness, and kurtosis can be calculated to capture statistical characteristics

of the data. Time-domain and frequency-domain features are also commonly used.

In some cases, domain experts or researchers with expertise in the specific activity domain may contribute to the feature extraction process [13]. They can provide insights into which aspects of the sensor data are most relevant for distinguishing activities.

Not all calculated or derived features may be equally informative for activity recognition. Feature selection is the process of identifying the most relevant features while discarding redundant or less informative ones. The choice of features and feature extraction techniques depends on the nature of the sensor data.

The selected features serve as a compact representation of the sensor data, highlighting the most salient information for activity recognition. These features may be organized as feature vectors, where each feature corresponds to a specific aspect of the data. Feature vectors are then used as input to machine learning models.

The dimensionality of the feature space has a significant impact on the requirements of a system, in particular for the real monitoring ones.

- **Training Data Requirement**: As the dimensionality of the feature space increases, a larger amount of training data is typically required to effectively estimate the parameters of models, such as machine learning classifiers. This is because higher-dimensional spaces have more complex decision boundaries.

- **Overfitting**: In high-dimensional feature spaces, there is a risk of overfitting, where the model performs well on the training data and performs poorly on unseen data. Adequate training data is essential to mitigate overfitting and ensure that the model can generalize on different activities.

- **Computational Intensity**: Working with high-dimensional feature spaces can be computationally intensive. Training and deploying recognition models in such spaces may require more processing power and memory capacity. This can be a challenge, especially for real-time or resource-constrained applications.

- **Curse of Dimensionality**: The curse of dimensionality refers to the exponential growth in data sparsity as the dimensionality of the feature space increases. This sparsity can make it challenging to find representative training examples and can lead to increased computational complexity.

2.2.4.7 Classification

Once the models are trained they have to predict the human activities being performed. The system assigns labels to activities based on the patterns it has learned during training.

In the field of Human Activity Recognition, several machine learning models have been applied to achieve accurate and robust activity recognition. Here's an overview of some commonly used machine learning algorithms in HAR:

- **K-Nearest Neighbors (K-NN)**: K-NN is a simple and effective algorithm used for classification tasks in HAR. It classifies activities based on the majority class among the k-nearest neighbors of a data point in feature space. It's known for its simplicity and ability to handle various types of data.

- **Decision Trees**: Decision trees are a popular choice for HAR due to their interpretability and ease of visualization. They work by recursively splitting the dataset based on features to create a tree-like structure for classification. Decision trees are prone to overfitting, but techniques like pruning can mitigate this issue.

- **Random Forest**: Random Forest is an ensemble learning method that combines multiple decision trees to improve classification accuracy and reduce overfitting. It works by aggregating the results of individual decision trees. Random Forest is known for its robustness and ability to handle large feature sets.

- **Linear Perceptron**: Linear Perceptron is a simple neural network model used for binary classification tasks. In HAR, it can be used for

distinguishing between two classes of activities, such as walking and running. While it's less complex than deep learning models, it can still be effective for certain HAR applications.

- **AdaBoost**: AdaBoost is an ensemble method that combines multiple weak classifiers to create a strong classifier. It iteratively adjusts the weights of misclassified samples to focus on difficult-to-classify instances. AdaBoost can be applied to improve the accuracy of HAR models.

- **Gaussian Naive Bayes**: Naive Bayes is a probabilistic classifier based on Bayes' theorem. Gaussian Naive Bayes assumes that features are normally distributed, making it suitable for continuous data, such as sensor readings in HAR. It's computationally efficient and can handle high-dimensional data.

Additionally, distinct deep learning models have been deployed in this research. Here, we present an overview of some frequently employed deep learning architectures in the field of HAR:

- **Fully connected layer**
- **Convolutional layer**
- **Batch Normalization layer**
- **Long Short-term Memory**
- **Attention layer**
- **Residual Learning**

2.2.4.7.1 Fully connected layer

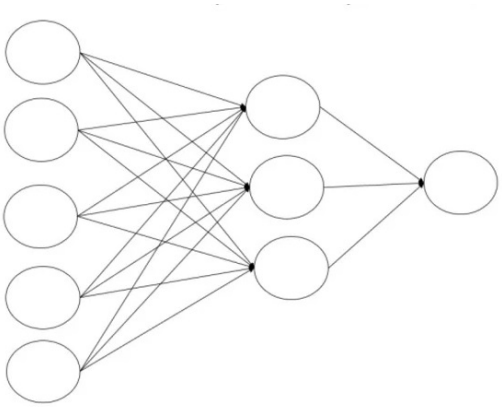


Figure 2.4: Enter Caption

The Fully Connected layer, often referred to as the Dense layer, represents the most straightforward neural network architecture. It establishes complete connections between every input neuron and each neuron within the dense layer. These connections execute a weighted sum with a bias term, and the outcome is subjected to an activation function to introduce non-linearity, as expressed by the following formula:

$$y = \sigma(W \cdot x + b) \tag{2.1}$$

Fully connected layers are able to capture relationships within the data and to extract hidden features.

2.2.4.7.2 Convolutional layer

The Convolutional layer is a fundamental building block of Convolutional Neural Networks (CNNs) and is specifically designed to capture high-level hidden features or patterns within data. Unlike dense layers, convolutional layers are sparsely connected to a small portion of the input data, which aids in local feature detection and helps reduce the number of trainable parameters. The core operation of these layers is known as convolution, which involves a weighted sum of the input data passed through a filter (also referred to as a kernel) treated as a vector of weights. In practical

terms, convolution operates using a sliding window approach, where the kernel sldes on the input data.

It's important to note that the dimension of the kernel dictates the dimension of the convolution operation. Assuming a continuous 2D signal, such as an image, a 2D convolution (represented as a matrix kernel) is defined as follows:

$$(i * h)[x, y] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} i[k, l] \cdot h[x - k, y - l]$$

While the term 'convolutional layer' might initially imply a direct application of convolution, as traditionally understood in mathematical contexts, it's crucial to clarify that in deep learning field, these layers execute correlation operations. This distinction derives from the fact that they do not involve the flipping of kernels because it's computationally more efficient.. The correlation formula is the following:

$$(i \circ h)[x, y] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} i[k, l] \cdot h[x + k, y + l]$$

So, while the formulas may appear identical, the practical application of whether the kernel is flipped or not distinguishes convolution from correlation in signal processing and deep learning contexts.

Still some properties hold to keep equivalences between correlation and convolution:

$$(i * h) = (h * i) = (h \circ i)$$

To speed up the MAD (multiplication and addition) operation computed in convolutional layers, separabilty can be applied. The key advantage of separable convolutions is that they break down a 2D convolution into two 1D convolutions. The convolution formula can the be rewritten as follow:

$$(i_s * h_x * h_y)[x, y] = \sum_{k=-\infty}^{\infty} h_x[x - k] \sum_{l=-\infty}^{\infty} i[k, l] \cdot h_y[y - l]$$

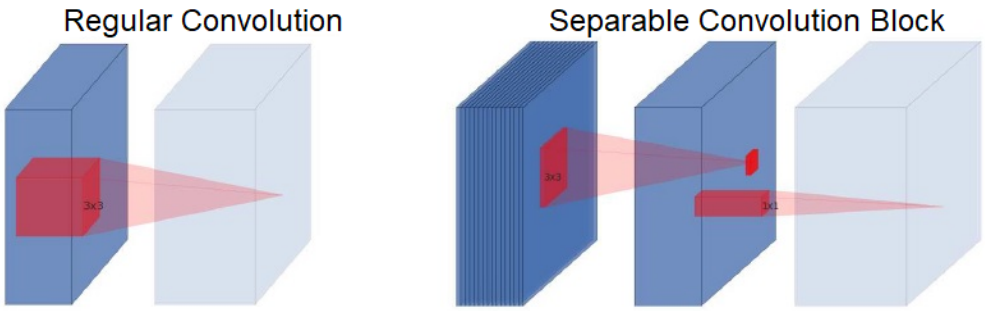


Figure 2.5: Graphical visualization of regular convolution and separable convolution

2.2.4.7.3 Batch Normalization layer

Batch Normalization is a technique designed to mitigate the issues caused by internal covariate shift. During the training phase, Batch Normalization normalizes the input data across the batch dimension. This process computes the mean and variance of the input data along the batch dimension. The input data is first flattened, then normalized with a mean of 0 and a variance of 1. Additional parameters are added allowing the network to learn the appropriate scaling and shifting of the original input activation values.

During the testing or inference phase of a neural network, the batch parameters used in Batch Normalization are not updated. During testing, the network operates on individual data samples, and there's no concept of mini-batches. Therefore, the mean and variance statistics remain fixed at the values computed during training. This ensures that the network's behavior is consistent and deterministic during inference.

Training Phase:

1. Compute the mean and variance over the mini-batch:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

2. Normalize the inputs within the mini-batch:

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$$

3. Scale and shift the normalized inputs:

$$y_i = \gamma\hat{x}_i + \beta$$

4. Update the moving average of mean and variance for later inference:

$$\mu_{\text{moving}} = \text{momentum} \cdot \mu_{\text{moving}} + (1 - \text{momentum}) \cdot \mu_B$$

$$\sigma^2_{\text{moving}} = \text{momentum} \cdot \sigma^2_{\text{moving}} + (1 - \text{momentum}) \cdot \sigma_B^2$$

Inference Phase:

1. During inference, the precomputed values (mean, variance, gamma, and beta) from the training phase are used. The normalization step is the same as in the training phase:

$$\hat{x}_i = \frac{x_i - \mu_{\text{moving}}}{\sqrt{\sigma^2_{\text{moving}} + \epsilon}}$$

2. And then the scaled and shifted output is computed:

$$y_i = \gamma\hat{x}_i + \beta$$

2.2.4.7.4 Long Short-term Memory

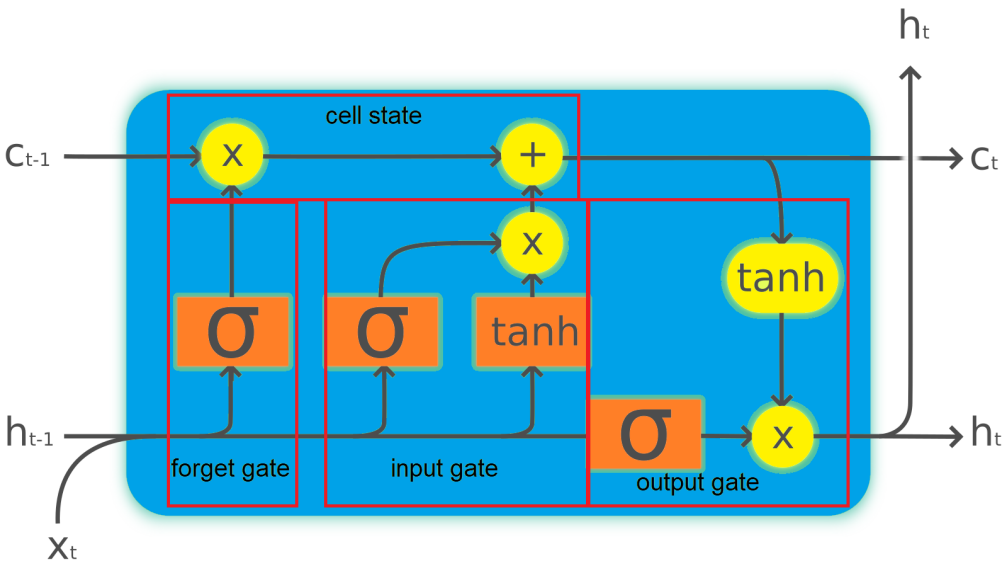


Figure 2.6: LSTM

The Long Short-term Memory (LSTM) layer plays an imporatl role in sensor-based Human Activity Recognition (HAR) due to its adeptness in learning intricate patterns and dependencies within sequential time series data. Functioning as a part of Recurrent Neural Network (RNN) modules, LSTMs are proficient in analyzing sequential data and have the capacity to capture long-term dependencies. They achieve this by maintaining an internal memory that can selectively retain or forget information as they process input sequences.

LSTMs operate sequentially on the input sequence, iteratively updating their memory. The number of time steps in this process is determined by the dimension of the input sequence, which also dictates the number of outputs generated by the layer (refer to Fig 2.6). Key components of an LSTM include:

- **Forget gate:** This gate regulates what content (comprising both previous computations and new information) needs to be forgotten. It does this by calculating the sigmoid of a weighted sum of inputs.

- **Input gate:** Responsible for regulating the flow of previous information that should be combined with the new content. This gate, also referred to as the update gate, calculates both a sigmoid gate and a candidate cell state.
- **Cell state:** The cell state is an internal long-term memory that tracks portions of previous and current content information.
- **Output gate:** This gate controls the flow of output information and calculates the output at each time step, forwarding the new content information to the next time step.

In sequential time series a bidirectional approach could be more powerful, thus leading to the creation of Bidirectional LSTMs (BiLSTMs). In a bidirectional LSTM, two LSTMs operate operate together but in different directions. One LSTM analyzes past dependencies from the beginning of the sequence, while the second LSTM operates in reverse, examining future information starting from the end of the sequence. These outputs are concatenated at each time step.

2.2.4.7.5 Attention layer

The Attention layer [14], primarily employed in Natural Language Processing (NLP) for tasks like Neural Machine Translation (NMT [15]), is known for its ability to perform sequential reasoning by selectively focusing on certain components while considering others. Although it's commonly used in NLP, this architecture is not limited to linguistic tasks and can be effectively applied to input sequences unrelated to language processing. In the context of HAR, the attention mechanism can help extract temporal dependencies from time series data.

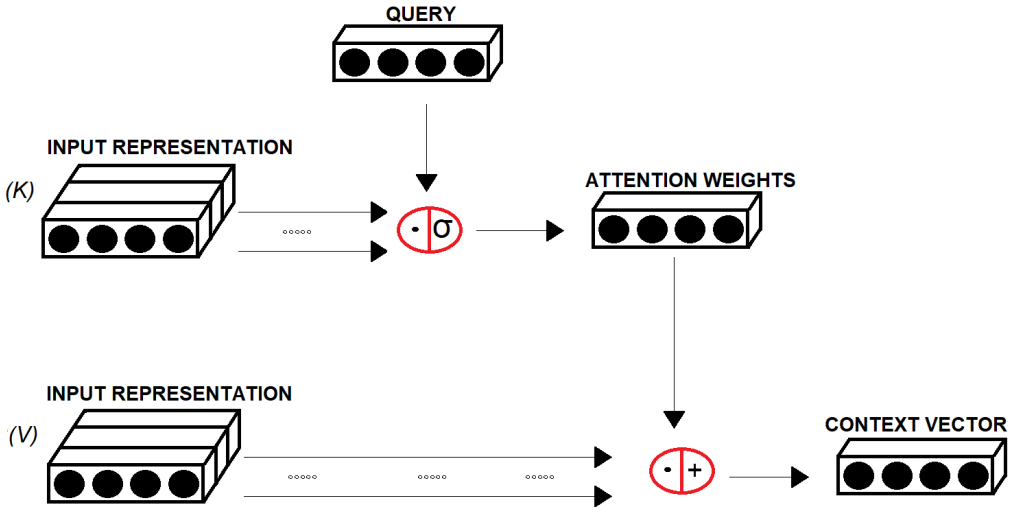


Figure 2.7: Attention Core

The attention mechanism (refer to 2.7) involves two key inputs: the context representation (K) and the query (q). These inputs are used to calculate a dot-product, followed by a softmax activation function, resulting in attention weights. These weights essentially represent the relevance of each input element to the query.

In some cases, a second context representation (V) may be necessary. To accommodate this, the attention mechanism can be generalized by incorporating the previous computation (referred to as the attention core) and introducing the second context representation (V). This architecture enables a weighted sum operation on the second representation using the attention weights, ultimately producing a context vector. This enhanced model excels at extracting meaningful features and relationships among different representations of the same context, even when K and V are identical.

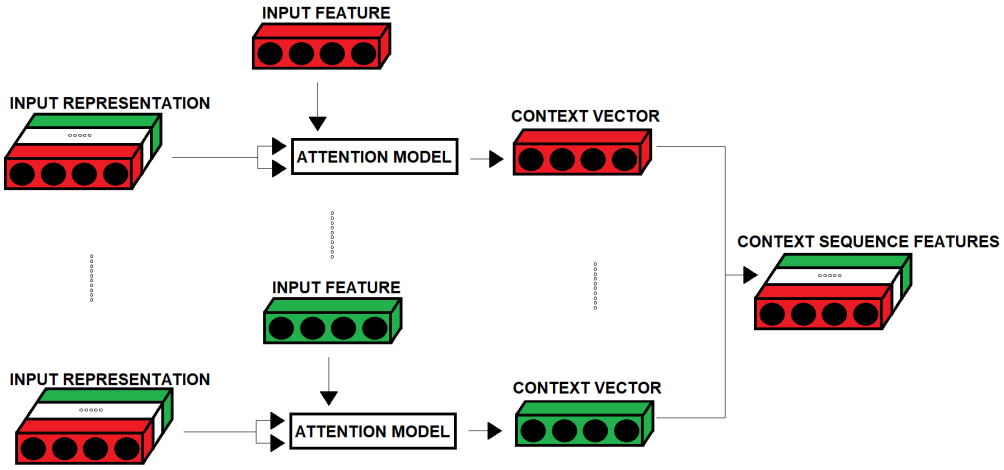


Figure 2.8: Self Attention Model

Furthermore, this attention model can be applied to extract features and relationships among individual elements of the same input sequence (refer to 2.8). This approach, known as self-attention, involves computing multiple attention mechanisms using the input sequence as context (K and V) and each sequence component as the query. In the context of this study, self-attention is employed to improve the extraction of time-relevant information among window elements in the HAR application.

2.2.4.7.6 Residual Learning

Residual Learning, often referred to as ResNet, is a powerful architectural innovation that has made a significant impact on deep learning and neural network training. Originally developed for image classification tasks, ResNet introduced the concept of residual blocks, which enable the training of very deep neural networks while mitigating the vanishing gradient problem. Although initially designed for computer vision tasks, the concept of residual learning can be applied to various domains, including HAR.

In the context of HAR, residual blocks can enhance the learning of complex temporal dependencies within time series data. By introducing skip connections, or shortcuts, that allow gradients to flow directly through a

block, ResNet enables the training of deeper networks without encountering diminishing gradient issues. This is particularly advantageous when working with HAR, as it involves the analysis of sequential data with intricate patterns and dependencies.

The key idea behind residual learning is the residual mapping, which aims to fit a residual rather than the actual desired mapping. This residual is learned during training and then added to the input data to obtain the final output. This approach allows the network to focus on learning the difference between the input and the desired output, making it easier to optimize deep architectures.

In summary, the adoption of residual learning in HAR can lead to more effective feature extraction and better modeling of temporal relationships within the data. This architecture has the potential to improve the accuracy and efficiency of activity recognition systems, making them more robust in real-world scenarios.

2.2.5 Evaluation

Evaluating the performance of an Activity Recognition Chain (ARC) is a critical aspect. Activity recognition systems can encounter various types of classification errors, including missing activities, confusing activities, or falsely detecting activities that did not occur. These errors are commonly categorized as follows:

- **True Positives (TPs)**: Correctly recognized and classified activities.
- **True Negatives (TNs)**: Correctly identified instances where activities did not occur.
- **False Positives (FPs)**: Incorrectly detected activities that did not occur.
- **False Negatives (FNs)**: Missed or incorrectly unrecognized activities.

The objective of optimization can vary depending on the application. In some cases, it may be more favorable to reduce FNs.

To assess the performance of ARC systems, several performance metrics are commonly used, including:

- **Accuracy**: Measures the overall correctness of activity classifications.
- **Precision**: Measures the ratio of correctly identified positive instances (TPs) to all instances classified as positive (TPs + FPs).
- **Recall**: Measures the ratio of correctly identified positive instances (TPs) to all actual positive instances (TPs + FNs).
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two metrics.
- **Confusion Matrix**: A tabular representation of classification results, showing the counts of TP, TN, FP, and FN.

The choice of which metric to optimize depends on the specific application and its requirements.

Chapter 3

Human Activity Recognition

3.1 Introduction

3.2 Contributions

This research was bring on by me and my colleague and friend Francesco Palmisano. This research was based one Zarmina Ursino thesis [16].

Pair programming, a methodology well-known in software development. It involves two researchers, myself and my colleague collaborating closely on every aspect of the project. One of the defining characteristics of our research process was the equitable sharing of responsibilities. We engaged in regular discussions and brainstorming sessions. We both made equal contributions to every aspect of our research, from the conception of hypotheses to the experimental methodologies. These dialogues were necessary in the direction our research would take. In fact, as a team, we faced challenges head-on, looking for the best approaches and solutions.

In conclusion, our collaborative approach significantly contributed to the success of our thesis project.

3.3 eSteps insoles

eSteps Inc., an American company with its roots in Bologna, Italy, was founded in 2020 under the leadership of Nidhal Louhichi, the current CEO. The company is known for the creation of cutting-edge insoles designed to capture and record data containing walking and running patterns. However, eSteps Inc. is currently focused in the creation of systems that can help the monitoring of Multiple Sclerosis. Today, through the help of smart insoles., eSteps Inc. creates a direct connection between MS patients and their medical practitioners.

Figure 3.1: eSteps insoles

These insoles facilitates the flow of information from the patient to the doctor. This exchange of data provides a huge amount of data that can be used to further improve the clinical picture of the patient. In each insole lies an IMU device, MetaMotionS, embedded into a commercially insole.

3.3.1 Sensor specification

MetaMotionS is in fact a wearable device that offers real-time and continuous monitoring of motion and environmental sensor data. What makes

MetaMotionS stand out is the integration of sensors and a rechargeable battery (70-100mAH) in a compact form of 27mm × 27mm x 4mm.

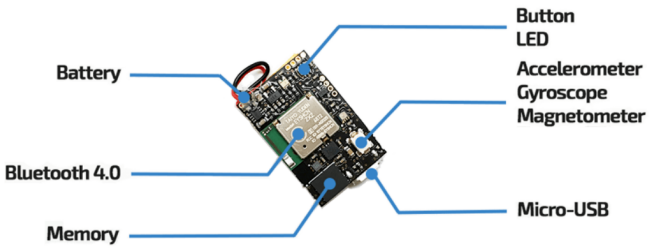


Figure 3.2: MetaMotionS components

The sensors utilized in our research are the gyroscope and the accelerometer, both operating at a high sampling rate of 100Hz. The gyroscope, with a precision of 2000 degrees per second, gives accurate measurements of angular velocity, while the accelerometer, with a sensitivity of 16g (16 times the acceleration due to gravity), provides precise information on linear acceleration along three axes (x, y, and z). The sensor is powered by a reliable 3.7 V lithium-ion battery with a capacity of 100mAH, making it easy to replenish using a standard micro-USB cable. This battery last for approximately 2 weeks on a single charge. An energy-saving feature of this sensor is the sleep mode functionality, which automatically activates when the sensor remains stationary, preserving battery life efficiently.

Gyroscope	Range:	± 125, ± 250, ± 500, ± 1000, ± 2000°/s
	Resolution:	16 bit
	Sample Rate:	0.001Hz – 100Hz stream – 800Hz log
Accelerometer	Range:	± 2, ± 4, ± 8, ± 16 g
	Resolution:	16 bit
	Sample Rate:	0.001Hz – 100Hz stream – 800Hz log
Magnetometer	Range:	±1300µT (x,y-axis), ±2500µT (z-axis)
	Resolution:	0.3µT
	Sample Rate:	0.001Hz – 25Hz
Barometer	Range:	300 – 1100 hPa
	Resolution:	0.01 hPa (< 10 cm)
	Accuracy:	~ ±1 hPa (abs), ± 0.12 hPa (±1 m – rel)
	Sample Rate:	0.001Hz – 50Hz
Sensor Fusion	Outputs:	Quaternion, Euler Angles (Yaw, Pitch, Roll), Linear Acc Earth Acc (Gravity), Robust Heading
	Accuracy:	<1° RMS
	Sample Rate:	100Hz
Temperature	Range:	-40°C – 85°C
	Resolution:	16 bit
	Sample Rate:	0.001Hz – 50Hz

Figure 3.3: MetaMotionS spec.

3.4 Data acquisition

In this study, data were collected from a diverse group of 14 participants, with various genders and ages, with a majority (70%) being male and the remaining (30%) female, falling within the age range of 18 to 70 years. The recorded activities consists of different kind of motions, including walking, brisk walking, running, ascending stairs, descending stairs, and moments of no movement.

Notably, the data acquisition process was conducted separately for each lower limb, offering several advantages. Firstly, it enabled a detailed acquisition, allowing for the analysis of individual gait steps. Secondly, this approach exhibited robustness regardless of of posture changes, as it primarily focused on the dynamics of gait steps, resulting in a more accurate method for detecting various types of motions.

During the data acquisition, participants followed a predefined protocol (ref to Fig. 3.4) designed to capture different activities within single recordings. The protocol involved the following steps:

- Initiating the recording from a standing position.
- Performing the targeted activity for a duration of 10 to 30 seconds.
- Maintaining a standing position for 10 seconds, signifying moments of inactivity.
- Repeating the activity for 10 seconds.
- This cycle continued until the recording was stopped.

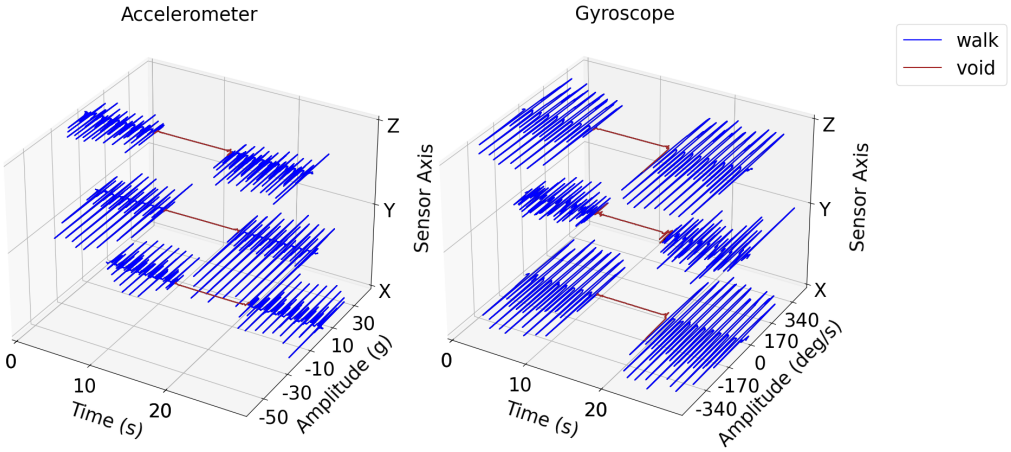


Figure 3.4: Walking recording

A crucial part of this research focuses on the creation of predictive models capable of accurately categorizing a wide range of human activities. The success of this classification task deeply depends on how individuals carry out these activities.

The data were collected in smartphones devices which were connected via Bluetooth.

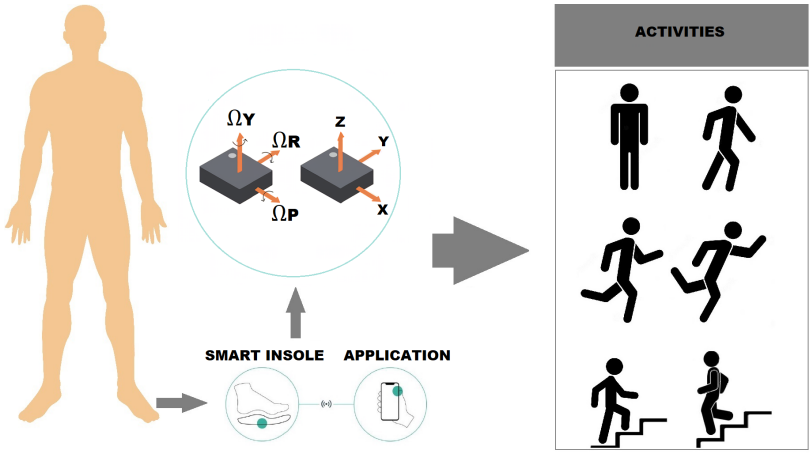


Figure 3.5: Data acquisition

According to Gait Analysis principles, each activity follows the structure of the gait cycle. Despite these activities sharing a common structure, they exhibit variations in terms of style, behavior, and the duration of gait cycles. Collecting data from a diverse range of individuals is very important since each person has a unique style or pattern in performing activities. The significance of gathering a diverse participant pool is a crucial aspect. In particular, even though activities share a common structural foundation, the way they are executed can differ significantly from one person to another. To capture this rich diversity and create robust predictive models, we collected data from individuals of various ages, genders, and backgrounds. By doing so, we ensure that our models not only accurately classify activities but also account for the inherent variability in human motion across different

individuals.

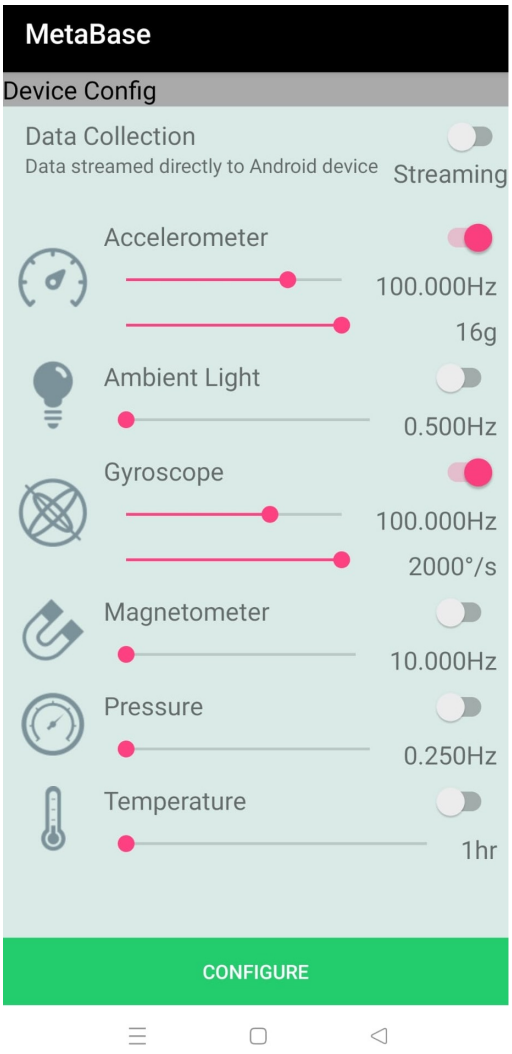


Figure 3.6: MetaBase configuration

Participants were requested to repeat the same activity multiple times, resulting in a rich repository of data points for each activity category. This repetition helped not only to increase the amount of data but also allowed

the model to notice small differences in how each person did the activities. By doing the same thing over and over, our participants unknowingly helped the model learn and understand their unique ways of moving. This careful approach made the model better at recognizing activities, giving more accurate results that match each person's particular style.

Furthermore, a different data acquisition (ref to Fig. 3.7) was done by performing specific activities for a duration of 10 seconds, followed by brief 10-second pauses before transitioning into a new activity.

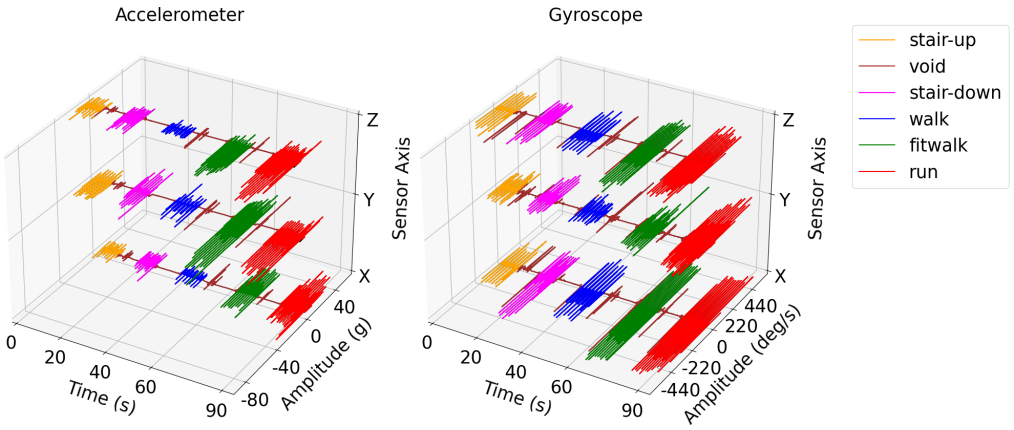


Figure 3.7: Complete path consisting of: walking, brisk walking, running, ascending stairs, descending stairs

3.5 Preprocessing

3.5.0.1 Data Cleaning

Once the data were acquired from the sensors and collected by the smartphone app, the next phase involved data refinement. Initially, the time series generated through the acquisition protocol underwent interpolation to address any gaps and missing data, ensuring uniformity in the sample frequency.

Following this, a low-pass filter was applied to the accelerometer and gyroscope signals. Our approach involves the implementation of a digital low-

pass filter to effectively separate the AC (Alternating Current) component from the DC (Direct Current) component within each time series of sensor data. In the context of HAR, this separation is crucial, as it helps distinguish dynamic motion-related activities (such as walking or running) from the influence of gravity. To achieve this, we set the filter's cutoff frequency at 0.25 Hz and 0.125Hz for gyroscope and accelerometer respectively [17]. By applying this low-pass filter, we are able to isolate the gravitational component and the body acceleration component along the three axes. This separation is achieved by subtracting the low-pass filtered data from the original signal along each axis. The result is distinct data streams representing gravitational acceleration and body acceleration in different directions. Our primary interest lies in body motion acceleration. Therefore, we select the AC components.

3.5.0.2 Data labeling

Accurate labeling of activities in our dataset is very important. Labels help us make sense of the raw sensor data and understand human activity patterns. Precise labeling not only adds meaning to the data but also enables us to build and test robust machine learning and deep learning models. When we define the boundaries of different activities with precision, we are able to distinguish between various movements effectively. Moreover, these labels are essential for assessing how well our models perform, ensuring that our activity recognition system functions at its best. In summary, the quality and accuracy of our labeled data have a direct impact on the success of our research, making labeling a vital part of our methodology.

Labeling datasets can be a difficult and time-consuming task, one that cannot be fully automated, as it requires human intervention to ensure accuracy. In the context of HAR, which deals with multiple activities within the same recording session, a significant challenge is delineating data points that demarcate activity regions accurately. This research proposes a semi-automatic algorithm designed to enhance the control and precision of activity labeling, offering a more efficient alternative to entirely manual annotation. The algorithm draws inspiration from a proprietary Matlab function, "findchangepts,". When re-implemented, this function returns

a list of indices of possible points within the input signal that mark a significant jump between different values. The most suitable metric to identify the jumps in gait cycles is the Root Mean Square function. The algorithm analyses the input signal and looks for any notable deviations in behavior. If the residual error surpasses a predetermined threshold, it indicates a change in the input signal's behavior within that window. To manage the complexity of the process, parameter such as the quantity of point to analyse and simplier metrics are available. The threshold parameter controls the quantity of change points produced as output. Lower thresholds yield more change points, while higher thresholds reduce their number.

While this function shows promise in detecting change points, it does come with certain limitations. Firstly, it may not identify the precise change points for every signal. It necessitate parameter tuning for each signal data recorded. Secondly, the time complexity of this function grows with very long signals. By choosing the right amount of signal point to analyse, a trade off between time complexity and accuracy can be set. The fewer points the algorithm analyses the faster the computation.

Accelerometer and gyroscope data are collected simultaneously, ensuring that the moments when an activity changes are synchronized between the two sensors. Consequently, the "findchangepts" algorithm was applied to the accelerometer data only. This decision was made based on the fact that the change points identified by the algorithm for the accelerometer data correspond directly to those for the gyroscope data, thanks to their simultaneous recording.

3.5.0.3 Data balancing

Balancing the data is a crucial step in the process of preparing a dataset for activity recognition. Data balancing ensures that each class or activity category within the dataset has an equal number of examples or instances.

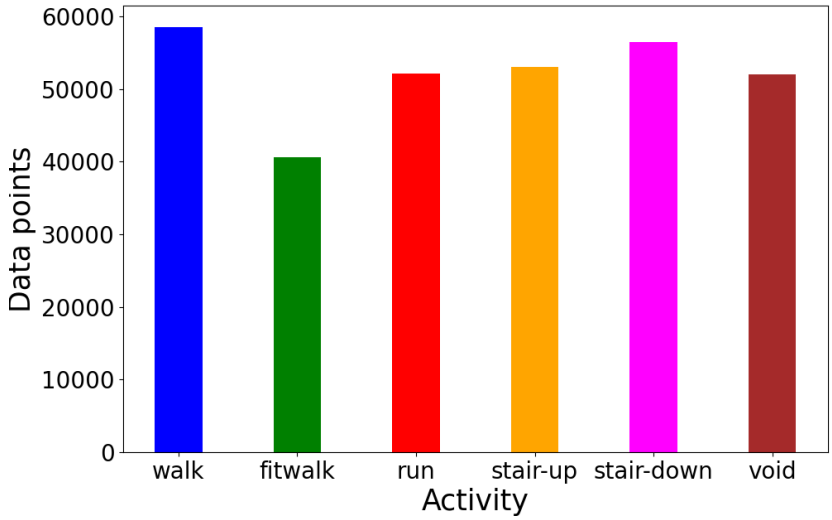


Figure 3.8: Number of point acquired on left foot

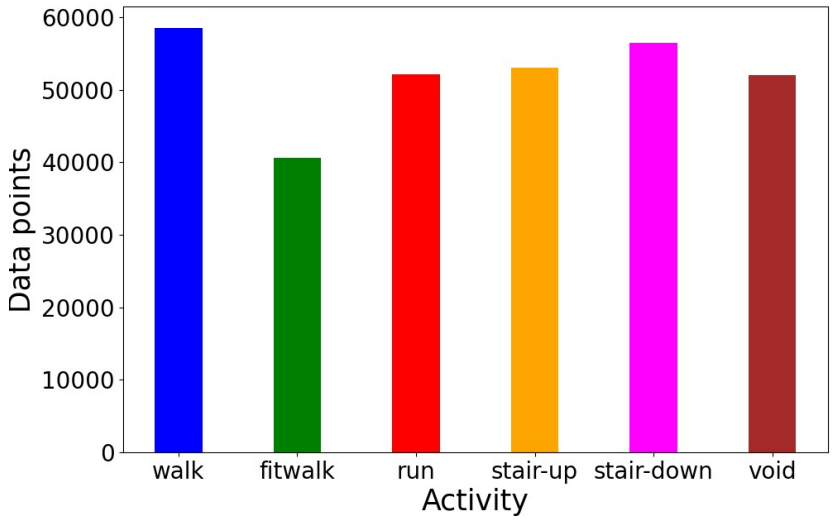


Figure 3.9: Number of point acquired on right foot

3.6 Data segmentation

The next step is preparing the dataset set for the Machine learning and deep learning models. To achieve this, a time-based windowing technique, often referred to as a sliding window, was implemented for segmenting the gathered data. Initially, a sliding window approach was employed within each activity region, as depicted in figure 3.10.

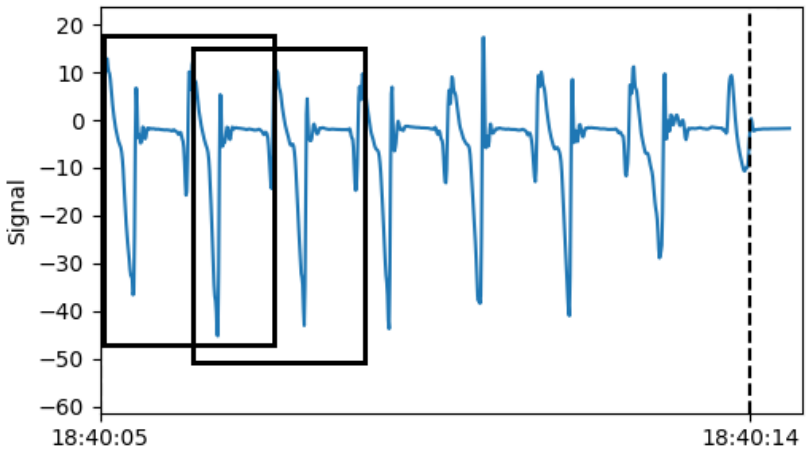


Figure 3.10: Sliding window

This approach allows for a more detailed analysis by capturing temporal dynamics within the dataset. By sliding the window through the dataset, subsequences were extracted, each containing distinct temporal segments.

To manage computational complexity effectively, we made strategic choices regarding window sizes, namely 100, 200, 300, 400, 700, 800, and 900. For each of these window sizes, a sliding shift was applied as a percentage of the respective window size, specifically at intervals of 10%, 30%, 50%, 70%, and 100%.

3.7 Feature extraction

Feature extraction is a crucial step in optimizing the performance of machine learning models, especially when dealing with high-dimensional data. Differently from deep learning models that can learn hidden features, feature extraction remains indispensable. In particular, is crucial to deal with the curse of dimensionality during the training of ML models.

As the window size and the number of spatial axes and sensors increase, the cardinality of inputs grows significantly. To effectively manage this issue, Fourier Transformations were applied to each window for every spatial axis, revealing the frequency domain characteristics embedded within the data. This transformation retained only the real values, discarding the imaginary component.

The resultant features were selected to significantly reduce dimensionality as weel as keeping crucial information within the data.

- **Mean Value**: This feature calculates the average value of the raw data vector, providing an indicator of central tendency.
- **Standard Deviation**: Calculating the standard deviation of the raw vector gives insights into data stability.
- **Median**: The median of the transformed vector offers a robust measure of central tendency, less sensitive to outliers.
- **Lower Quartile**: This feature denotes the 25th percentile of the transformed vector, it delineates the lower range of data distribution.
- **Upper Quartile**: In contrast to the lower quartile, the upper quartile represents the 75th percentile, highlighting the upper range of data distribution.
- **Skewness**: Skewness measures the asymmetry of the transformed vector's distribution, showing its shape and symmetry.
- **Kurtosis**: The kurtosis of the transformed vector provides insights into the distribution's peakedness or flatness.

3.8 Classification

In this research, all the models employ an inverted bottleneck structure inspired by MobileNetV2 [18]. This structure consists of two Inverted Residual (ID) separable convolutional layers. The first layer has 32 kernels, followed by the second layer with 12 kernels, both with a dimension of one. Following each convolutional layer, there is a batch normalization layer to stabilize the distribution of hidden features. After the inverted bottleneck structure, there is a series of fully connected (FC) layers with a decreasing number of units. These layers, called also dense layers, use the ReLU activation function, except for the last layer, which employs the softmax activation function to produce the output probability distribution. To prevent overfitting, each dense layer (except the last one) incorporates a dropout process before activation. The ResNet architecture is applied to the input data and integrated into the output of the bottleneck layer. This integration is a common approach in deep neural networks, where residual connections or skip connections are used to facilitate the flow of gradients during training and enable the network to learn more effectively.

The models used in this research differ in the content of the bottleneck. The baseline model (refer to Fig. 3.11) contains only two sequential dense layers. In contrast, the second model (refer to Fig. 3.12) incorporates a Self-Attention architecture followed by a Bidirectional Long Short-Term Memory (BiLSTM) layer, repeated twice. Self-Attention helps capture temporal relations and patterns among hidden features, while BiLSTM is utilized to analyze complex temporal sequences. The third model (refer to Fig. 3.13) consists of a single dense layer followed by a Self-Attention layer with BiLSTM.

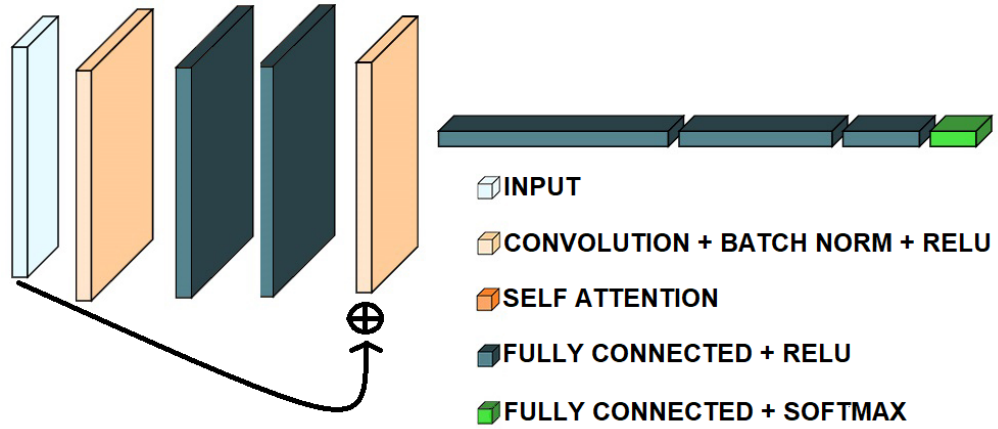


Figure 3.11: ResnNet Baseline: 2xFC + Self Attention

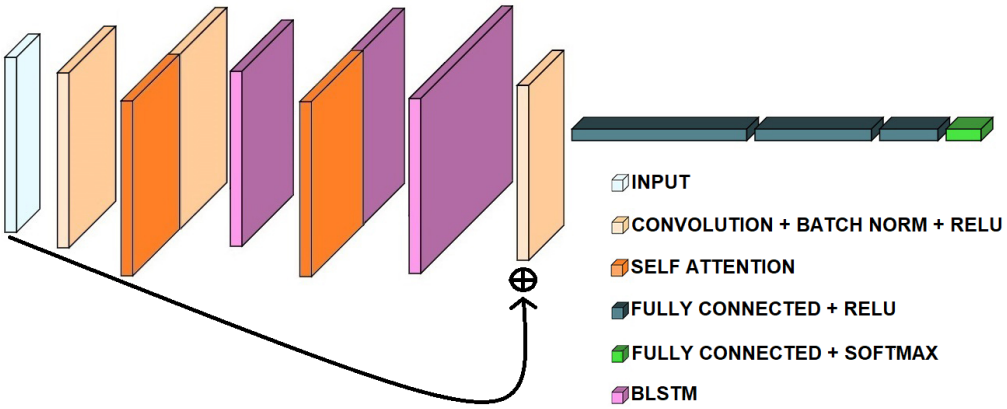


Figure 3.12: ResnNet 2xBiLSTM + Self Attention

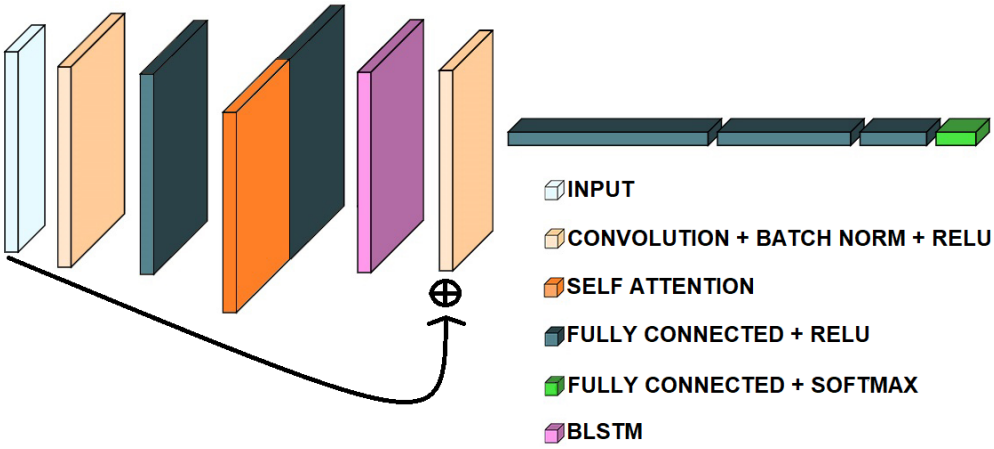


Figure 3.13: ResnNet FC + BiLSTM + Self Attention

3.9 Evaluation

3.9.1 Robustness

In the context of HAR, achieving robust and accurate model performance is very important. The conventional k-fold cross-validation approach is a widely used technique for assessing model performance, where the dataset is divided into subsets or "folds" for training and testing. However, this approach often has limitations to the subject-specific characteristics present in the dataset. This can lead to models that are not robust when faced with entirely new subjects during testing.

In many HAR scenarios, individuals may exhibit unique styles or variations in how they perform certain activities. When evaluating a HAR model's effectiveness, it's essential to account for these variations to ensure that the model can generalize well beyond the subjects used for training.

This is where Leave One Subject Out Cross-Validation (LOSOCV [19]) comes into play. LOSOCV is a specialized variant of k-fold cross-validation that addresses the subject-specificity issue. Instead of dividing the dataset into arbitrary folds, LOSOCV divides it into as many folds as there are

individual subjects in the dataset. Each fold, in this case, consists exclusively of data from a single subject.

By organizing the dataset in this manner, LOSOCV ensures that the model's performance is rigorously evaluated for each subject independently. This approach allows for a more finer assessment of how well the model can adapt to and recognize activities performed by different individuals.

In practical terms, LOSOCV provides a more realistic simulation of how the HAR model will perform in the real world. It acknowledges that people have distinct ways of performing activities, and the model must be robust enough to handle these variations. By treating each subject as a separate fold, LOSOCV allows the model to generalize across a diverse range of individuals, regardless of the unique characteristics of different subjects..

Following the tuning process of all models with the LOSOCV methodology, our analysis moved into assessing the robustness of each model through an average accuracy calculation for each subject. This metric was chosen as it provides a reliable indicator of overall performance.

3.9.2 Evaluation analysis

Our testing approach is composed of two distinct analyses: evaluation and inference, by assessing both retrospective and real-time usage scenarios.

3.9.2.1 Deep learning models

Category	Model	Window	Sliding	Accuracy	Patient mean accuracy	Precision weighted	Recall weighted	F1-Score weighted
DL	ResNet Baseline	100	10%	0.94	83.49	0.94	0.94	0.94
DL	ResNet Baseline	200	10%	0.99	86.49	0.99	0.99	0.99
DL	ResNet Baseline	300	10%	0.95	88.09	0.95	0.95	0.95
DL	**ResNet Baseline**	**400**	**10%**	**0.97**	**88.74**	**0.97**	**0.97**	**0.97**
DL	ResNet Baseline	700	10%	0.97	87.04	0.97	0.97	0.97
DL	ResNet Baseline	800	10%	0.96	83.78	0.96	0.96	0.96
DL	ResNet Baseline	900	10%	0.96	79.21	0.96	0.96	0.95
DL	ResNet 2xBiLSTM + SelfAttention	100	10%	0.96	84.84	0.96	0.96	0.96
DL	ResNet 2xBiLSTM + SelfAttention	200	10%	0.96	87.0	0.96	0.96	0.96
DL	ResNet 2xBiLSTM + SelfAttention	300	10%	0.94	87.77	0.94	0.94	0.94
DL	**ResNet 2xBiLSTM + SelfAttention**	**400**	**10%**	**0.97**	**86.48**	**0.98**	**0.97**	**0.97**
DL	ResNet 2xBiLSTM + SelfAttention	700	10%	0.97	83.06	0.97	0.97	0.97
DL	ResNet 2xBiLSTM + SelfAttention	800	10%	0.95	78.62	0.95	0.95	0.95
DL	ResNet 2xBiLSTM + SelfAttention	900	10%	0.94	71.3	0.95	0.94	0.94
DL	ResNet FC + BiLSTM + SelfAttention	100	10%	0.97	85.36	0.97	0.97	0.97
DL	ResNet FC + BiLSTM + SelfAttention	200	10%	0.96	87.19	0.97	0.96	0.97
DL	ResNet FC + BiLSTM + SelfAttention	300	10%	0.94	87.96	0.94	0.94	0.94
DL	**ResNet FC + BiLSTM + SelfAttention**	**400**	**10%**	**0.98**	**88.52**	**0.98**	**0.98**	**0.98**
DL	ResNet FC + BiLSTM + SelfAttention	700	10%	0.98	85.99	0.98	0.98	0.98
DL	ResNet FC + BiLSTM + SelfAttention	800	10%	0.96	80.89	0.97	0.96	0.96
DL	ResNet FC + BiLSTM + SelfAttention	900	10%	0.95	74.23	0.95	0.95	0.95

Figure 3.14: Evaluation of Deep Learning models on left limb

In the evaluation analysis, we assessed the quality of the models by considering precision, recall, and F1-Score metrics for each test subject, along with assessing the robustness through the mean accuracy of subjects. The analysis revolved around activity regions categorized by their labels. We observed that as the window size was around 400 milliseconds, the quality of results reached the best performance. This trend indicated that models benefited from having access to context nor too small nor too large or more information surrounding the input data during the evaluation process. Indeed, by accessing to even greater window sizes the performances slowly worsen.

Examining the confusion matrix of the best model (Resnet with fully connected layers and BiLSTM)for the largest window size (refer to Fig. 3.14), it's evident that the model almost perfectly classified long-range activities, segmented by the region detector during feature extraction, achieving 98% F1-Score and accuracy. The same model with the smallest window got 97% accuracy and F1-Score. Additionally, the accuracy mean obtained from each subject utilized in the LOSOCV approach ranged from 81% to 89% for the best model, further highlighting the robustness of our approach.

Category	Model	Window	Sliding	Accuracy	Patient mean accuracy	Precision weighted	Recall weighted	F1-Score weighted
DL	ResNet Baseline	100	10%	0.9	80.71	0.91	0.9	0.9
DL	ResNet Baseline	200	10%	0.9	83.2	0.9	0.9	0.9
DL	ResNet Baseline	300	10%	0.94	83.93	0.95	0.94	0.94
DL	**ResNet Baseline**	**400**	**10%**	**0.93**	**84.3**	**0.94**	**0.93**	**0.94**
DL	ResNet Baseline	700	10%	0.93	79.51	0.94	0.93	0.93
DL	ResNet Baseline	800	10%	0.91	76.37	0.92	0.91	0.91
DL	ResNet Baseline	900	10%	0.92	72.33	0.94	0.92	0.93
DL	ResNet 2xBiLSTM + SelfAttention	100	10%	0.87	82.96	0.88	0.87	0.87
DL	ResNet 2xBiLSTM + SelfAttention	200	10%	0.92	85.03	0.93	0.92	0.92
DL	ResNet 2xBiLSTM + SelfAttention	300	10%	0.91	86.19	0.92	0.91	0.91
DL	**ResNet 2xBiLSTM + SelfAttention**	**400**	**10%**	**0.94**	**86.49**	**0.94**	**0.94**	**0.94**
DL	ResNet 2xBiLSTM + SelfAttention	700	10%	0.93	80.15	0.94	0.93	0.93
DL	ResNet 2xBiLSTM + SelfAttention	800	10%	0.93	73.97	0.94	0.93	0.93
DL	ResNet 2xBiLSTM + SelfAttention	900	10%	0.91	68.26	0.92	0.91	0.91
DL	ResNet FC + BiLSTM + SelfAttention	100	10%	0.94	83.93	0.94	0.94	0.94
DL	ResNet FC + BiLSTM + SelfAttention	200	10%	0.93	85.8	0.94	0.93	0.93
DL	ResNet FC + BiLSTM + SelfAttention	300	10%	0.94	86.72	0.94	0.94	0.94
DL	**ResNet FC + BiLSTM + SelfAttention**	**400**	**10%**	**0.96**	**87.5**	**0.97**	**0.96**	**0.96**
DL	ResNet FC + BiLSTM + SelfAttention	700	10%	0.95	82.46	0.96	0.95	0.95
DL	ResNet FC + BiLSTM + SelfAttention	800	10%	0.92	75.94	0.94	0.92	0.93
DL	ResNet FC + BiLSTM + SelfAttention	900	10%	0.93	70.2	0.94	0.93	0.93

Figure 3.15: Evaluation of Deep Learning models on right limb

The performance on the right foot (refer to Fig. 3.15) closely mirrors that of the left foot, differing by just a minor 1-point variation. This similarity highlights the consistency in the data collected from both limbs, enhancing the overall reliability of our study's results.

3.9.2.2 Machine learning models

Category	Model	Window	Sliding	Accuracy	Patient mean accuracy	Precision weighted	Recall weighted	F1-Score weighted
ML	Gaussian Naive Bayes	100	30%	0.67	66.04	0.68	0.67	0.67
ML	Gaussian Naive Bayes	200	100%	0.75	71.96	0.76	0.75	0.75
ML	Gaussian Naive Bayes	300	70%	0.78	72.67	0.79	0.78	0.78
ML	Gaussian Naive Bayes	400	100%	0.8	73.98	0.82	0.8	0.8
ML	Gaussian Naive Bayes	700	50%	0.79	74.78	0.8	0.79	0.79
ML	**Gaussian Naive Bayes**	**800**	**30%**	**0.79**	**76.36**	**0.79**	**0.79**	**0.79**
ML	Gaussian Naive Bayes	900	100%	0.81	73.79	0.82	0.81	0.81
ML	**Linear Perceptron**	**100**	**10%**	**0.59**	**51.97**	**0.62**	**0.59**	**0.55**
ML	Linear Perceptron	200	10%	0.52	46.33	0.61	0.52	0.49
ML	Linear Perceptron	300	10%	0.53	47.06	0.67	0.53	0.49
ML	Linear Perceptron	400	30%	0.57	43.29	0.62	0.57	0.53
ML	Linear Perceptron	700	100%	0.49	42.45	0.56	0.49	0.48
ML	Linear Perceptron	800	30%	0.48	41.38	0.65	0.48	0.46
ML	Linear Perceptron	900	10%	0.6	37.07	0.7	0.6	0.58
ML	Random Forest	100	10%	0.96	74.73	0.97	0.96	0.96
ML	Random Forest	200	30%	0.97	77.99	0.97	0.97	0.97
ML	Random Forest	300	100%	0.97	79.36	0.97	0.97	0.97
ML	Random Forest	400	100%	0.97	80.82	0.97	0.97	0.97
ML	Random Forest	700	50%	0.97	82.89	0.97	0.97	0.96
ML	**Random Forest**	**800**	**100%**	**0.97**	**83.82**	**0.97**	**0.97**	**0.97**
ML	Random Forest	900	70%	0.96	83.1	0.96	0.96	0.96

Figure 3.16: Evaluation of Machine Learning models on left limb

The results (refer to Fig. 3.16) consistently showed that the models tend to perform better with larger window sizes. This observation suggests that providing a broader context or more information surrounding the input data during the evaluation process, enhances the models' ability to make accurate predictions. The increased window size allows the models to consider more extensive information and context, leading to improved performance, especially during the evaluation phase. A larger window enabled the model to consider a wider context, potentially leading to superior performance on the evaluation dataset. This highlight the importance of context in accurately recognizing and classifying human activities, emphasizing the utility of larger window sizes in achieving better model performance.

Category	Model	Window	Sliding	Accuracy	Patient mean accuracy	Precision weighted	Recall weighted	F1-Score weighted
ML	Gaussian Naive Bayes	100	10%	0.64	65.08	0.65	0.64	0.64
ML	Gaussian Naive Bayes	200	50%	0.74	71.93	0.74	0.74	0.73
ML	Gaussian Naive Bayes	300	100%	0.74	70.93	0.75	0.74	0.74
ML	Gaussian Naive Bayes	400	50%	0.75	74.26	0.77	0.75	0.75
ML	Gaussian Naive Bayes	700	100%	0.77	73.39	0.79	0.77	0.77
ML	**Gaussian Naive Bayes**	**800**	**100%**	**0.8**	**76.15**	**0.81**	**0.8**	**0.8**
ML	Gaussian Naive Bayes	900	70%	0.75	71.14	0.76	0.75	0.74
ML	Linear Perceptron	100	50%	0.51	42.09	0.56	0.51	0.49
ML	**Linear Perceptron**	**200**	**10%**	**0.53**	**45.54**	**0.68**	**0.53**	**0.48**
ML	Linear Perceptron	300	30%	0.55	42.39	0.63	0.55	0.55
ML	Linear Perceptron	400	10%	0.5	44.17	0.6	0.5	0.44
ML	Linear Perceptron	700	70%	0.56	39.07	0.54	0.56	0.5
ML	Linear Perceptron	800	30%	0.57	40.91	0.58	0.57	0.56
ML	Linear Perceptron	900	50%	0.46	38.17	0.52	0.46	0.4
ML	Random Forest	100	70%	0.96	72.11	0.96	0.96	0.96
ML	Random Forest	200	50%	0.98	74.75	0.98	0.98	0.98
ML	Random Forest	300	100%	0.97	74.12	0.97	0.97	0.97
ML	Random Forest	400	100%	0.98	76.01	0.98	0.98	0.98
ML	**Random Forest**	**700**	**100%**	**0.98**	**78.03**	**0.98**	**0.98**	**0.98**
ML	Random Forest	800	100%	0.98	77.61	0.98	0.98	0.98
ML	Random Forest	900	30%	0.97	75.9	0.97	0.97	0.97

Figure 3.17: Evaluation of Machine Learning models on right limb

Similarly to what we observed for deep learning models with the right foot, the performance of machine learning models doesn't exhibit significant differences between the two limbs – left and right (refer to Fig. 3.17). There's only a marginal one-point variation in their performance. This consistency in performance between the left and right limbs assesses the robustness of our models in recognizing and classifying human activities, regardless of whether they are performed with the left or right foot. It further validates the effectiveness of our approach in achieving balanced and accurate results across both limbs.

3.9.3 Inference analysis

Category	Model	Window	Sliding	Accuracy	Precision weighted	Recall weighted	F1-Score weighted
DL	**ResNet Baseline**	**100**	**10%**	**0.82**	**0.93**	**0.82**	**0.86**
DL	ResNet Baseline	200	10%	0.75	0.94	0.75	0.83
DL	ResNet Baseline	300	10%	0.71	0.98	0.71	0.82
DL	ResNet Baseline	400	10%	0.61	0.96	0.61	0.74
DL	ResNet Baseline	700	10%	0.35	0.92	0.35	0.5
DL	ResNet Baseline	800	10%	0.28	0.95	0.28	0.43
DL	ResNet Baseline	900	10%	0.24	0.95	0.24	0.38
DL	**ResNet 2xBiLSTM + SelfAttention**	**100**	**10%**	**0.83**	**0.94**	**0.83**	**0.88**
DL	ResNet 2xBiLSTM + SelfAttention	200	10%	0.76	0.95	0.76	0.84
DL	ResNet 2xBiLSTM + SelfAttention	300	10%	0.71	0.98	0.71	0.82
DL	ResNet 2xBiLSTM + SelfAttention	400	10%	0.64	0.99	0.64	0.77
DL	ResNet 2xBiLSTM + SelfAttention	700	10%	0.35	0.93	0.35	0.5
DL	ResNet 2xBiLSTM + SelfAttention	800	10%	0.31	0.96	0.31	0.46
DL	ResNet 2xBiLSTM + SelfAttention	900	10%	0.24	0.91	0.24	0.37
DL	**ResNet FC + BiLSTM + SelfAttention**	**100**	**10%**	**0.82**	**0.93**	**0.82**	**0.87**
DL	ResNet FC + BiLSTM + SelfAttention	200	10%	0.75	0.93	0.75	0.83
DL	ResNet FC + BiLSTM + SelfAttention	300	10%	0.67	0.94	0.67	0.78
DL	ResNet FC + BiLSTM + SelfAttention	400	10%	0.61	0.96	0.61	0.74
DL	ResNet FC + BiLSTM + SelfAttention	700	10%	0.35	0.94	0.35	0.51
DL	ResNet FC + BiLSTM + SelfAttention	800	10%	0.31	0.95	0.31	0.46
DL	ResNet FC + BiLSTM + SelfAttention	900	10%	0.24	0.93	0.24	0.37

Figure 3.19: Inference of Deep Learning models on left limb

In the inference analysis, we evaluated models by simulating real-time predictions made sequentially on the test data. The results revealed that both deep and machine learning models tended to perform better with smaller window sizes. This observation suggests that models benefits from considering less context when making predictions on new data (as shown in Fig. 3.19 3.20 3.21 3.22). Smaller context windows allowed the model to focus on specific details pertinent to the input, such as the gait cycle, leading to more accurate real-world predictions.

Category	Model	Window	Sliding	Accuracy	Precision weighted	Recall weighted	F1-Score weighted
DL	**ResNet Baseline**	**100**	**10%**	**0.78**	**0.88**	**0.78**	**0.82**
DL	ResNet Baseline	200	10%	0.73	0.92	0.73	0.81
DL	ResNet Baseline	300	10%	0.67	0.95	0.67	0.78
DL	ResNet Baseline	400	10%	0.58	0.96	0.58	0.73
DL	ResNet Baseline	700	10%	0.31	0.89	0.31	0.46
DL	ResNet Baseline	800	10%	0.23	0.88	0.23	0.37
DL	ResNet Baseline	900	10%	0.19	0.88	0.19	0.31
DL	**ResNet 2xBiLSTM + SelfAttention**	**100**	**10%**	**0.83**	**0.93**	**0.83**	**0.88**
DL	ResNet 2xBiLSTM + SelfAttention	200	10%	0.75	0.93	0.75	0.83
DL	ResNet 2xBiLSTM + SelfAttention	300	10%	0.67	0.94	0.67	0.79
DL	ResNet 2xBiLSTM + SelfAttention	400	10%	0.6	0.96	0.6	0.74
DL	ResNet 2xBiLSTM + SelfAttention	700	10%	0.32	0.92	0.32	0.47
DL	ResNet 2xBiLSTM + SelfAttention	800	10%	0.24	0.9	0.24	0.37
DL	ResNet 2xBiLSTM + SelfAttention	900	10%	0.19	0.79	0.19	0.3
DL	**ResNet FC + BiLSTM + SelfAttention**	**100**	**10%**	**0.82**	**0.91**	**0.82**	**0.86**
DL	ResNet FC + BiLSTM + SelfAttention	200	10%	0.75	0.94	0.75	0.83
DL	ResNet FC + BiLSTM + SelfAttention	300	10%	0.68	0.95	0.68	0.79
DL	ResNet FC + BiLSTM + SelfAttention	400	10%	0.6	0.97	0.6	0.74
DL	ResNet FC + BiLSTM + SelfAttention	700	10%	0.32	0.94	0.32	0.48
DL	ResNet FC + BiLSTM + SelfAttention	800	10%	0.25	0.84	0.25	0.38
DL	ResNet FC + BiLSTM + SelfAttention	900	10%	0.17	0.71	0.17	0.27

Figure 3.20: Inference of Deep Learning models on right limb

This phenomenon can be attributed to the fact that smaller window sizes enabled the model to capture finer details at boundary points, where classification errors often occurred. In cases where an activity transitioned directly into another (boundary activity) or when a single window included multiple activities, larger windows made it challenging for the model to discern precise boundaries, resulting in ambiguous predictions.

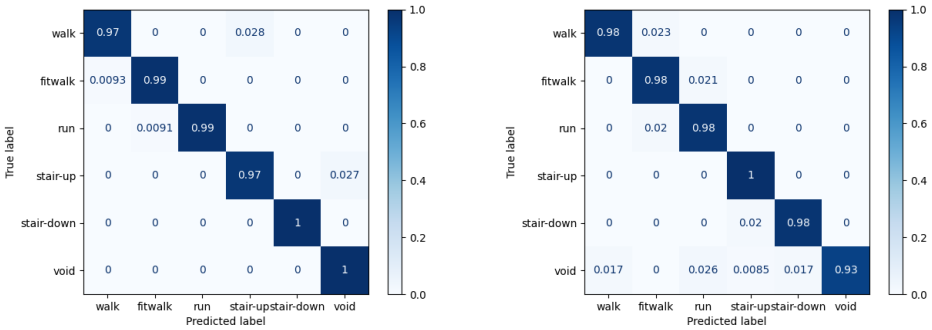
Category	Model	Window	Sliding	Accuracy	Precision weighted	Recall weighted	F1-Score weighted
ML	Gaussian Naive Bayes	100	30%	0.75	0.82	0.75	0.78
ML	Gaussian Naive Bayes	200	100%	0.8	0.83	0.8	0.81
ML	**Gaussian Naive Bayes**	**300**	**70%**	**0.84**	**0.88**	**0.84**	**0.85**
ML	Gaussian Naive Bayes	400	100%	0.81	0.83	0.81	0.81
ML	Gaussian Naive Bayes	700	50%	0.63	0.86	0.63	0.71
ML	Gaussian Naive Bayes	800	30%	0.5	0.95	0.5	0.64
ML	Gaussian Naive Bayes	900	100%	0.72	0.76	0.72	0.71
ML	**Linear Perceptron**	**100**	**10%**	**0.61**	**0.66**	**0.61**	**0.62**
ML	Linear Perceptron	200	10%	0.4	0.66	0.4	0.44
ML	Linear Perceptron	300	10%	0.51	0.7	0.51	0.56
ML	Linear Perceptron	400	30%	0.37	0.71	0.37	0.38
ML	Linear Perceptron	700	100%	0.53	0.64	0.53	0.54
ML	Linear Perceptron	800	30%	0.35	0.6	0.35	0.42
ML	Linear Perceptron	900	10%	0.12	0.77	0.12	0.18
ML	**Random Forest**	**100**	**10%**	**0.81**	**0.9**	**0.81**	**0.85**
ML	Random Forest	200	30%	0.8	0.91	0.8	0.85
ML	Random Forest	300	100%	0.85	0.86	0.85	0.85
ML	Random Forest	400	100%	0.84	0.84	0.84	0.84
ML	Random Forest	700	50%	0.66	0.88	0.66	0.73
ML	Random Forest	800	100%	0.73	0.75	0.73	0.73
ML	Random Forest	900	70%	0.8	0.83	0.8	0.81

Figure 3.21: Inference of Machine Learning models on left limb

After comprehensive analysis across multiple window sizes, it was determined that windows between 1 and 2 seconds are an optimal balance between the evaluation and inference phases. The confusion matrix of the best model and window sizes indicated that the lowest result on the main diagonal was associated with the void activity, as each activity was consistently preceded and followed by a standing position (void activity).

Category	Model	Window	Sliding	Accuracy	Precision weighted	Recall weighted	F1-Score weighted
ML	Gaussian Naive Bayes	100	10%	0.73	0.83	0.73	0.77
ML	**Gaussian Naive Bayes**	**200**	**50%**	**0.82**	**0.89**	**0.82**	**0.85**
ML	Gaussian Naive Bayes	300	100%	0.82	0.84	0.82	0.82
ML	Gaussian Naive Bayes	400	50%	0.78	0.89	0.78	0.83
ML	Gaussian Naive Bayes	700	100%	0.77	0.78	0.77	0.76
ML	Gaussian Naive Bayes	800	100%	0.7	0.74	0.7	0.69
ML	Gaussian Naive Bayes	900	70%	0.78	0.83	0.78	0.8
ML	**Linear Perceptron**	**100**	**50%**	**0.64**	**0.76**	**0.64**	**0.65**
ML	Linear Perceptron	200	10%	0.52	0.77	0.52	0.57
ML	Linear Perceptron	300	30%	0.41	0.69	0.41	0.45
ML	Linear Perceptron	400	10%	0.29	0.69	0.29	0.36
ML	Linear Perceptron	700	70%	0.33	0.57	0.33	0.27
ML	Linear Perceptron	800	30%	0.29	0.7	0.29	0.38
ML	Linear Perceptron	900	50%	0.44	0.61	0.44	0.47
ML	**Random Forest**	**100**	**70%**	**0.85**	**0.88**	**0.85**	**0.86**
ML	Random Forest	200	50%	0.8	0.88	0.8	0.83
ML	Random Forest	300	100%	0.78	0.81	0.78	0.78
ML	Random Forest	400	100%	0.8	0.82	0.8	0.8
ML	Random Forest	700	100%	0.69	0.72	0.69	0.7
ML	Random Forest	800	100%	0.7	0.71	0.7	0.7
ML	Random Forest	900	30%	0.35	0.84	0.35	0.47

Figure 3.22: Inference of Machine Learning models on right limb



(a) Evaluation of best model on left foot (b) Evaluation of best model on right foot

(c) Inference of best model on left foot (d) Inference of best model on right foot

Figure 3.18: Confusion matrices

Chapter 4

Conclusions

4.1 Conclusions

This research studies in deep the task of classifying human activities through the utilization of sensory data acquired via intelligent insoles. The dataset utilized here consists of recordings obtained from a diverse group of healthy subjects, each following distinct paths during the data collection process. To prepare this data, a rigorous sequence of data cleaning, segmentation, and feature extraction was performed. This systematic approach was designed to enable machine learning to extract information and discern recurring patterns.

In contrast, deep learning models were modelled to autonomously identify and extract features directly from the raw input data. This methodology eliminated the need for manual feature extraction. In the initial evaluation phase, the model delivered remarkable results, with a mean accuracy rate of 89%, with a F1-Score of 98%. In the inference phase, the model continued to demonstrate its robustness, maintaining a commendable accuracy rate of 84%, supported by a F1-Score of 88%.

What became increasingly apparent was the influence of window size on model performance. A window sizes of 400 milliseconds substantially improved evaluation performance, whereas smaller windows proved invaluable in addressing boundary prediction challenges encountered during the inference phase. The optimal window size was determined to fall within the

range of 1 to 2 seconds, striking an effective balance between evaluation accuracy and inference robustness. —— As a potential avenue for future research, expanding the dataset to include a more diverse array of subjects, characterized by varying attributes and activity styles, holds the promise of further enhancing the model's performance in real-world scenarios. This study underscores the immense potential of intelligent insoles coupled with advanced machine learning techniques in revolutionizing the field of human activity recognition. It sets the stage for continued exploration and innovation in this dynamic domain.

4.2 Future works

This research serves as a solid foundation upon which future research can be built, offering a multitude of promising directions for extension and expansion. There are several promising directions for future exploration and development:

- **Model Refinement**: A promising future step involves the creation of a fully automatic region detector, further automating and optimizing the activity recognition process.
- **Multimodal Fusion**: The integration of multiple sensor modalities, such as synchronized sensors for each foot, opens up new possibilities. Additionally, combining sensor-based data acquisition with vision-based approaches can enrich the dataset, necessitating the development of a front-end application for efficient and low-latency data synchronization.
- **Performance Enhancement**: Fine-tuning performance evaluations can lead to more robust models. Training on a more extensive dataset including a diverse range of patient characteristics could significantly improve system performance.
- **Multiple Sclerosis Monitoring**: Expanding the dataset to include data from multiple sclerosis patients would enable the analysis of

activity patterns in the context of a motor disease scenario, potentially offering valuable insights for disease monitoring and management.

- **Commercialization**: As this research lays the groundwork for practical applications, future efforts can focus on the commercialization of the product. Implementing a robust post-market monitoring system will be essential to monitor the product's performance and safety in real-world healthcare settings.

Exploring these directions for future research will contribute to the improvement of activity recognition systems. By doing so, we can enhance the practicality of these systems, making them even more valuable in real-world applications.

Acknowledgement

I would like to sincerely thank, first and foremost, my thesis advisor, Professor Andrea Asperti, and Nidhal Louhichi, eSteps' CEO, for making this thesis journey possible.

Heartfelt thanks to my family and parents for supporting me in every academic decision I've made so far.

Another thank you goes to my friends who have accompanied me on this journey.

A heartfelt thank you also goes to my colleague and friend, Francesco Palmisano, with whom I completed not only this thesis and research, but most of the project throughout our university career.

A special thank you is to all the students who, like me, are about to graduate.

And finally, thanks to the University of Bologna, without which I could not have undertaken this journey.

Bibliography

[1] Lynsey Lakin, Bryan E Davis, Cherie C Binns, Keisha M Currie, and Mary R Rensel. Comprehensive approach to management of multiple sclerosis: addressing invisible symptoms—a narrative review. *Neurology and therapy*, 10:75–98, 2021.

[2] Michael M. Goldenberg. Multiple sclerosis review. *P T*, 37(3):175–184, Mar 2012.

[3] Calabresi P. A. Diagnosis and management of multiple sclerosis. *American family physician*, 70(10):1935–1944, 2004.

[4] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):1–33, 2014.

[5] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du, and Duan-Sheng Chen. A comprehensive survey of vision-based human action recognition methods. *Sensors*, 19(5):1005, 2019.

[6] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[7] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[9] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 816–833. Springer, 2016.

[10] Alejandro Baldominos, Alejandro Cervantes, Yago Saez, and Pedro Isasi. A comparison of machine learning and deep learning techniques for activity recognition using mobile devices. *Sensors*, 19(3):521, 2019.

[11] Tsige Tadesse Alemayoh, Jae Hoon Lee, and Shingo Okamoto. New sensor data structuring for deeper feature extraction in human activity recognition. *Sensors*, 21(8):2814, 2021.

[12] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Hector Pomares, and Ignacio Rojas. Window size impact in human activity recognition. *Sensors*, 14(4):6474–6499, 2014.

[13] Mi Zhang and Alexander A Sawchuk. Motion primitive-based human activity recognition using a bag-of-features approach. In *Proceedings of the 2nd ACM SIGHIT international health informatics symposium*, pages 631–640, 2012.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[16] Zarmina Ursino. Recognition of human activities through esteps insoles. Master's thesis.

[17] Akram Bayat, Marc Pomplun, and Duc A Tran. A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, 34:450–457, 2014.

[18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[19] Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2521–2537, 2011.

List of Figures

THE END