Wearable sensor-based rehabilitation exercise assessment for post-stroke rehabilitation

Issam Boukhennoufa

BA, MS

Thesis Submitted for the Degree of Doctor of Philosophy

School of Computer Science and Electronic Engineering

University of Essex

February 2024

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Issam Boukhennoufa

February 2024

Acknowledgments

I would like to express my profound gratitude to my principal supervisor, Dr. Xiaojun Zhai, for his unwavering guidance, wisdom, and patience throughout my Ph.D. journey. Dr. Zhai not only provided invaluable mentorship but also offered me opportunities to shape my career. His support during personal challenges was invaluable, and his accessibility was a source of comfort.

I extend my sincere appreciation to my two other esteemed supervisors, Prof. Klaus D. McDonald-Maier and Dr. Victor Utti. Prof. McDonald-Maier consistently facilitated my professional growth, while Dr. Utti's assistance in gaining access to Colchester's hospital stroke unit for data collection and provision of invaluable insights into stroke rehabilitation were indispensable. I am also grateful to my colleagues at the School of Computer Science and Electronic Engineering at the University of Essex.

Furthermore, I wish to acknowledge Prof. Faycal Bensaali, without whom my Ph.D. journey would not have been possible. I extend my gratitude to Prof. Abbes Amira and my dear friend Dr. Hamza Djelouat for providing me with the opportunity to embark on a research career.

In addition to my academic mentors, I am indebted to my father, sister, brother, and wife for their unwavering support and encouragement throughout my Ph.D. journey. Lastly, I offer special recognition to my mother, whose lifelong encouragement and instilled appreciation for a career in science and the pursuit of excellence have been instrumental in my academic pursuits.

I dedicate this achievement to my beloved daughter, Lynn, whose presence has been a source of joy and motivation.

Abbreviations

Term	**Description**
AB	Able Bodied
ADL	Activities of Daily Living
ANNs	Artificial Neural Networks
ARAT	Action Research Arm Test
CAHAI	Chedoke Arm and Hand Activity Inventory
CNN	Convolutional Neural Network
COM	Center of Mass
CV	Computer Vision
DCT	Discrete Cosine Transform
DL	Deep Learning
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
EMG	Electromyography
FCN	Fully Convolutional Neural Networks
FMA	Fugi Mayer Assessment

FFT	Fast Fourier Transforms
GAN	Generative Adversarial Networks
GADF	Gramian Difference Angular Fields
GASF	Gramian Summation Angular Fields
GMAF	Gramian Angular Fields
GRF	Ground Reaction Force
HAR	Human Activity Recognition
IMU	Inertial Measurement Units
kNN	k-Nearest Neighbour
LCSS	Longest Common Sub-sequence
LR	Learning Rate
ML	Machine Learning
MKV	Markov Transition Fields
MLP	Multilayer Perceptron
NLP	Natural Language Processing
NIHSS	National Institutes of Health Stroke Scale
OGM	Oxford Grading Motor-Scale
PCA	Principal Component Analysis

PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PSD	Power Spectral Density
RF	Random Forest
RMSE	Root Mean Square Error
RMS	Root Mean Square
ROM	Range Of Motion
RT	Random Trees
SD	Standard Deviation
SN	Siamese Networks
SVM	Support Vector Machines
TIA	Transient Ischemic Attack
TS	Time Series
TS-SGAN	Time Series Siamese GAN
TUG	Timed Up and Go
VGG	Visual Geometry Group
WMFT	Wolf Motor Function Test

Publications

Journal papers:

1. Boukhennoufa, I., Jarchi, D., Zhai, X., Utti, V., Sanei, S., Lee, T. K., & McDonald-Maier, K. D. (2023). A novel model to generate heterogeneous and realistic time-series data for post-stroke rehabilitation assessment. IEEE Transactions on Neural Systems and Rehabilitation Engineering., vol. 31, pp. 2676-2687, 2023, doi: 10.1109/TNSRE.2023.3283045.

2. Boukhennoufa, I., Altai, Z., Zhai, X., Utti, V., McDonald-Maier, K. D., & Liew, B. X. (2022). Predicting the Internal Knee Abduction Impulse During Walking Using Deep Learning. Frontiers in Bioengineering and Biotechnology, 10., doi:10.3389/fbioe.2022.877347.

3. Boukhennoufa, I., Zhai, X., Utti, V., Jackson, J., & McDonald-Maier, K. D. (2022). Wearable sensors and machine learning in post-stroke rehabilitation assessment: A systematic review. Biomedical Signal Processing and Control, 71, 103197, doi:10.1016/j.bspc.2021.103197, doi: 10.1016/j.bspc.2021.103197.

4. Zhu X, Boukhennoufa I, Liew B, Gao C, Yu W, McDonald-Maier KD, Zhai X. Monocular 3D Human Pose Markerless Systems for Gait Assessment. Bioengineering (Basel). 2023 May 26;10(6):653. doi: 10.3390/bioengineering10060653.

5. Altai Z, Boukhennoufa I, Zhai X, Phillips A, Moran J, Liew BXW. Performance of multiple neural networks in predicting lower limb joint moments using wearable sensors. Front Bioeng Biotechnol. 2023 Jul 31;11:1215770. doi: 10.3389/fbioe.2023.1215770.

Conference papers:

1. Boukhennoufa, I., Zhai, X., Utti, V., Jackson, J., McDonald-Maier, K.D. (2022). Encoding Sensors' Data into Images to Improve the Activity Recognition in Post Stroke Rehabilitation Assessment. In: Pattern Recognition and Artificial Intelligence. ICPRAI 2022. Lecture Notes in Computer Science, vol 13364. Springer, Cham. doi: 10.1007/978-3-031-09282-4_10

2. Boukhennoufa I, Zhai X, Utti V, Jackson J, McDonald-Maier KD. A comprehensive evaluation of state-of-the-art time-series deep learning models for activity-

recognition in post-stroke rehabilitation assessment. Annu Int Conf IEEE Eng Med Biol Soc. 2021 Nov;2021:2242-2247. doi: 10.1109/EMBC46164.2021.9630462.

3. Boukhennoufa I, Zhai X, Utti V, Jackson J, McDonald-Maier KD. A comprehensive evaluation of state-of-the-art time-series deep learning models for activity-recognition in post-stroke rehabilitation assessment. Annu Int Conf IEEE Eng Med Biol Soc. 2021 Nov;2021:2242-2247. doi: 10.1109/EMBC46164.2021.9630462.

4. Zhu, X., Boukhennoufa, I., Liew, B., McDonald-Maier, K. D., & Zhai, X. (2022, September). A Kalman Filter based Approach for Markerless Pose Tracking and Assessment. In 2022 27th International Conference on Automation and Computing (ICAC) (pp. 1-7). IEEE doi: 10.1109/ICAC55051.2022.9911152.

Book chapter:

- Liew, B., Pizzocaro, S., Zhai, X., Galasso, S., Rugmar, D., Waterkeyn T., Boukhennoufa, I., Zhu, X., De Nunzio, A. Motion analysis in neurological rehabilitation: from the lab to the clinic.

Abstract

This thesis focuses on the use of wearable sensors (WS) and machine learning (ML) algorithms in post-stroke rehabilitation assessment. The conventional approach to rehabilitation involves subjective clinical assessments and frequent therapy sessions, which are time-consuming, costly, and often limited in availability. To address these limitations, WS have emerged as a portable and cost-effective solution, enabling patients to perform rehabilitation exercises at home. These sensors provide quantitative data on patients' movements, allowing for continuous monitoring and assessment. Additionally, ML algorithms offer the potential to enhance the accuracy and efficiency of rehabilitation assessment by processing the data collected from WS.

The research presented in this thesis first aims to analyse recent developments in WS-based post-stroke rehabilitation assessment, identify limitations in the field, and propose state-of-the-art ML algorithms to improve assessment performance. The primary motivation is to provide a more comprehensive, personalised, and objective evaluation of motor function and mobility, leading to improved rehabilitation outcomes and quality of life for stroke survivors.

Chapter 2 provides a comprehensive literature review that examines the current state-of-the-art in post-stroke rehabilitation assessment, specifically focusing on the utilisation of wearable sensors and machine learning techniques. The review encompasses a thorough examination of commonly

employed sensors, targeted body limbs, outcome measures, study designs, and machine learning approaches. Furthermore, the review highlights the limitations encountered by researchers in the field, particularly pertaining to the accuracy of assessment algorithms and the availability of data.

Subsequent chapters in this thesis address these identified limitations by proposing innovative solutions. Chapter 3 presents an approach aimed at enhancing the accuracy of assessment algorithms by adapting widely used computer vision algorithms to the time-series domain. This adaptation enables more precise and reliable analysis of the collected time-series data, thereby improving the assessment process.

In Chapter 4, a novel methodology is introduced, which involves the transformation of time-series data into images and the subsequent utilisation of computer vision algorithms for assessment purposes. Furthermore, a linear interpolation methodology is implemented to adjust the size of the encoded images, allowing for an increase or decrease in dimensions. A comprehensive comparative analysis is then conducted to evaluate the impact of image size on the performance of the assessment algorithm.

Finally, Chapter 5 introduces a novel algorithm that generates heterogeneous and realistic data, which serves to enhance the rehabilitation assessment process. By generating synthetic data that closely resembles real-world scenarios, this algorithm addresses the limitation of limited data availability, ultimately leading to more robust and accurate assessments.

The contributions of each chapter provide insights into the current state-of-the-art in WS-based rehabilitation assessment, algorithm optimisation, data encoding techniques, and data augmentation strategies. The findings of this research aim to advance post-stroke rehabilitation outcomes and contribute to a more accurate and personalised assessment for stroke survivors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

On a global scale, more than 13.7 million cases of stroke occur each year, and it is important to note that around one-quarter of individuals above the age of 25 will experience this health issue at some point in their lives [1]. A stroke is a brain attack that occurs when blood flow is cut off to a part of the brain, subsequently resulting in the death of brain cells [2, 3]. There are three main types of stroke [4]: Transient Ischemic Attack (TIA) [5], ischemic stroke [6], and hemorrhagic stroke [7].

1. TIA is caused by a temporary interruption to the blood supply to the brain and may result in no lasting neurological deficit, it is considered to be a precursor and warning of a future stroke.
2. Ischemic stroke which is estimated at 87 per cent of strokes [8], occurs when a blood vessel supplying blood to the brain is obstructed.
3. Hemorrhagic stroke happens when a blood vessel ruptures [9].

Brain damage caused by stroke - if not deadly - will influence how the body functions including instigating temporary or permanent paralysis [10, 11]. Subsequently, some stroke survivors will make a quick recovery, while others will need help and more time to recuperate, and relearn skills they lost [12, 13].

To speed up the process of recovery, and to regain their independence, post-stroke patients ought to engage in physical therapy or rehabilitation [14, 15]. The conventional approach is for physical therapists to evaluate the physical activities of patients through visual observation, clinical impression, or tests and measures [16–18]. Rehabilitation activities might include:

- Motor skill exercises: to ameliorate the strength of the muscles and body coordination [19].
- Mobility training: in order to relearn functional activities including walking which may include the use of, mobility aids, such as walkers, wheelchairs and canes to help support the body's weight [20].
- Constraint-induced rehabilitation or forced-use therapy: to improve limb function, where the patients practise using the affected limb while the unaffected one is held still [21].
- Active or passive Range Of Motion (ROM): to help patients regain the ROM of the affected body joints [22].

However, this approach presents many limitations [23], indeed the availability of therapy may be limited and the patients need regular consultations in order to achieve their goals [24], moreover the additional expense of public and private transport from and to hospitals are an additional burden to the patient's finances [25]. Also, transportation to hospitals may cause discomfort and pain to post-stroke patients who lack the mobility and energy to leave their houses and periodically visit their doctors for training sessions [26]. Besides, doctors and therapists are overwhelmed with the workload with sessions lasting more than half an hour - on average - with a cadence of many sessions per week [27].

To tackle these issues, researchers have developed applications to assess rehabilitation outcomes using novel technologies namely "Wearable Sensors" (WS) [28], which provide a high level of portability and low price giving researchers and therapists a plethora of possibilities and solutions [29]. Indeed, WS allow patients to execute their exercises at home relieving them of the drain of transportation. Subsequently, several types of sensing devices are used in applications extending from monitoring subjects' physiologic responses like Electromyography (EMG) [30], Electrocardiogram (ECG) [31], or glucose level in the blood [32] to evaluating kinematics of the individuals: gait, ROM, balance using Inertial Measurement Units (IMU) [33]. These sensors are employed in conjunction with clinical tests and outcome measures, such as sit-to-stand [34], Timed Up and Go (TUG) [35] to give an objective assessment and monitoring of the patient condition [36].

Besides, the breakthrough in Machine Learning (ML) that provides outstanding performance tasks that used to require a lot of knowledge and time to model [37], as well as the tremendous advances made in processing system technologies that made the ML computing possible have given researchers more tools and resources to handle and process the data collected from the sensors and hence permitting a more accurate and quicker assessment [38]. Figure 1.1 shows an example of a WS-based rehabilitation assessment step.

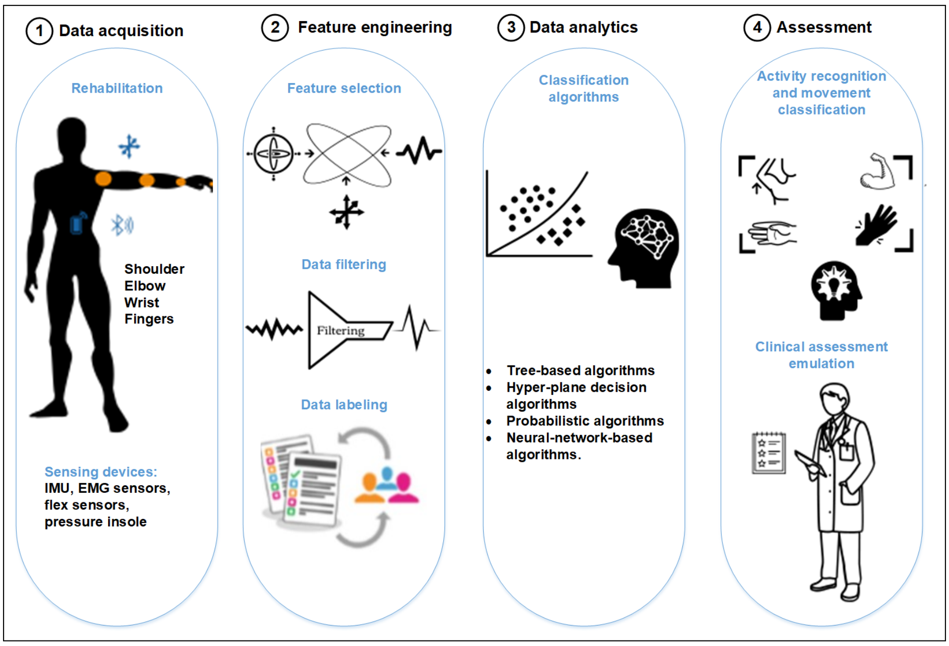


Figure 1.1: Wearable sensor-based rehabilitation assessment steps.

1.2 Research motivation

Conventional clinical assessments are frequently based on subjective interpretation of patient movements, leading to a time-consuming and often inaccurate evaluation of motor function and mobility. In contrast, WS can provide continuous and accurate data on a patient's movement patterns, allowing for a more comprehensive and objective assessment. Moreover, WS can facilitate unsupervised rehabilitation training, which can lead to improved patient outcomes and reduced healthcare costs.

In addition to these benefits, the use of ML algorithms in conjunction with WS has the potential to revolutionise post-stroke rehabilitation by providing more personalised and comprehensive assessments of motor function and mobility. This technology has the potential to enhance the quality of life of stroke survivors by facilitating faster and more effective rehabilitation outcomes, leading to improved long-term functional outcomes and overall health.

The primary aim of this thesis is to perform a comprehensive analysis of recent advancements in the field of WS for post-stroke rehabilitation. The research focuses on identifying the prevalent limitations, particularly those associated with the accuracy of assessment algorithms and the scarcity of available data. To address these challenges, state-of-the-art ML algorithms are proposed to enhance assessment performance as well as facilitate data augmentation to overcome the difficulties in collecting sufficient data in the field of post-stroke rehabilitation.

1.3 Contributions and dissertation structure

This section is structured as follows: Chapter 2 presents a comprehensive literature review of the current state-of-the-art in post-stroke rehabilitation assessment using WS and ML. Also, the existing limitations in the field are discussed. Chapter 3 proposes an approach to enhance the accuracy of assessment algorithms by adapting popular computer vision (CV) algorithms to the time series (TS) domain. Chapter 4 discusses a new methodology that encodes TS data into images and uses CV algorithms for the assessment. Chapter 5 introduces a novel algorithm to generate heterogeneous and realistic data to improve rehabilitation assessment.

A summary of each chapter's contribution is given below:

Chapter 2

The primary objective of this chapter is to conduct a comprehensive assessment of recent developments in the field of post-stroke rehabilitation utilising wearable devices for data collection and ML algorithms for exercise evaluation. To achieve this, a review was conducted. To categorise the assessment systems, a taxonomy was proposed that divided them into three categories: activity recognition, movement classification, and clinical assessment emulation. Additionally, this chapter provides a review of the most commonly utilised sensors, targeted body limbs, outcome measures, and study designs. Furthermore, the ML approaches utilised, starting from feature engineering to classification, are examined. Lastly, the limitations in the field are presented they were found to pertain to the accuracy and quantity of available data. The subsequent chapters of this thesis aim to provide viable solutions to address these.

Chapter 3

An important part of this chapter is to develop an efficient evaluation algorithm that provides a high-precision activity recognition rate in post-stroke rehabilitation assessment. Sixteen state-of-the-art TS deep learning (DL) algorithms with four different architectures were investigated: eight Convolutional Neural Networks (CNNs) configurations, six recurrent neural networks, a combination of the two and finally a wavelet-based neural network. Additionally, data from different sensors' combinations and placements as well as different pre-processing algorithms were explored to determine the optimal configuration for achieving the best performance. Our results show that the XceptionTime CNN architecture is the best-performing algorithm with normalised data. Moreover, it was found that sensor placement is the most important attribute to improve the accuracy of the system.

Chapter 4

A novel pipeline for TS classification is presented, it involves imaging the segmented TS data by employing three encoding techniques namely: Gramian Summation Angular Fields (GASF), Gramian Difference Angular Fields (GADF) and Markov Transition Fields (MKV). These encoding techniques were originally designed for univariate TS, one contribution of this work is to propose a way to adapt it to multivariate TS by imaging each axis of the sensors separately and fusing them to create multi-channel images. Another limitation comes from the fact that the resulting image size equals the sequence length of the original TS, this is tackled by employing a linear interpolation on the TS sequence to increase or decrease it. A comparison of the performance accuracy for the employed encoding technique and the image size has been done. Results showed that GASF and GADF performed better than MTF encoding, besides fusing the images and increasing the image size to a certain limit improved the accuracy from 83% for the ExceptionTime model to 91.5%. Finally, the proposed pipelines outperformed the existing stat-of-the-art accuracy on the same dataset by 4%. This pipeline represents a solution to the performance of the assessment algorithms identified in Chapter 2.

Chapter 5

One way to acquire more data is to use data augmentation that generates synthetic data by taking into account prior real data configuration. Generative Adversarial Networks (GANs) are one of the most recently used techniques. GANs have been found to suffer from mode collapse, which is an issue where the generated data did not take into account all the information from the original dataset. The objective of this paper is to tackle this problem.

To do so, a GAN is used to generate data for a real-world post-stroke clinical assessment dataset. As the original GAN was found to suffer from mode collapse, a new framework is proposed that involves adding a Siamese network (SN) and another discriminator to create Time Series Siamese GAN (TS-SGAN). Analysis using the longest common Subsequence (LCSS) showed that TS-SGAN created data uniformly for all the elements of the real dataset, contrary to the Original GAN. Moreover, encoding the generated dataset into images using Gramian Angular Field (GMAF) and classifying them using ResNet-18 allowed to improve the classification performance of an activity recognition dataset from 48.73% to 90.8% and from 63% to 98.2% for an ARAT dataset. This new model represents a solution to the data quantity limitation in post-stroke rehabilitation that was identified in Chapter 2.

Chapter 2

Literature review

2.1 Background on the study and review

The recent surge in technology-based stroke rehabilitation methods has facilitated the creation of effective rehabilitation settings, offering controlled and adaptable stimulation [39, 40]. These swift advancements have given rise to innovative approaches in stroke rehabilitation aimed at restoring motor functions in stroke survivors. Various interventions have been developed and assessed for stroke rehabilitation, including robot-assisted interventions [41–51], virtual reality [52–59], and WS-based approaches [60–95]. These interventions offer advantages such as task-specific and repetitive training, along with adaptive feedback, which enhances neuroplasticity and motor functions, thereby accelerating recovery. This thesis scope deals mainly with the application of WS-based rehabilitation so other approaches are not discussed.

The objective of this chapter is to evaluate the progress made in the domain of WS-based stroke rehabilitation assessment and to make a status report of the different technological developments in smart upper and lower limb recovery, to answer the following questions:

- What are the different aims of the post-stroke rehabilitation systems?
- What wearable sensing devices are more used?
- What are the most common outcome measures and the targeted sensors' place-

ments?

- What are the different study designs followed by the researcher in this field?
- Which ML algorithms and feature engineering techniques were more used?
- What limitations and challenges are encountered by researchers?

In the ensuing section, A comprehensive discussion of the relevant works in the field is presented, which includes an assessment of the WS utilised, the outcome measures, the types of assessment systems, and the diverse algorithms utilised. Finally, it entails a detailed exposition of the limitations and challenges encountered in post-stroke rehabilitation. Finally, the section concludes by providing suggestions for potential avenues to develop more effective systems.

2.2 Discussion about WS based rehabilitation

Study characteristics related to the WS used and its placement, the monitored exercises, the participants, the selected features the ML algorithm used and the classification performance for the included papers are presented in table A.1. The studies are divided into three categories based on the assessment type namely activity recognition, movement classification and clinical assessment emulation (explained below). After that, a more in-depth discussion on each topic is done separately with a quantitative comparison done at the end of this section.

2.2.1 Assessment systems and outcome measures

In the post-stroke rehabilitation, and based on the reviewed papers, a new taxonomy was gleaned in which the assessment systems in post-stroke rehabilitation were classified. Subsequently, three assessment approaches depending on the system's aim were distinguished: activity recognition, movement classification and clinical assessment emulation.

Activity recognition

Are systems which aim to identify specific movements of rehabilitation of the patients and differentiate between them for record and monitoring purposes [60–71], in this category researchers monitored Activities of Daily Living (ADL) [96] and they most frequently covered detecting general activities like standing, sitting, lying, standing up, sitting down [61, 63, 66, 67, 69], performing kitchen tasks like making a drink, chopping food [61] and other routine activities like making the bed, reading and lacing shoes [67], folding, sweeping and brushing teeth [65, 67, 68]. Other researchers covered activities for specific body parts like recognising different hand gestures [60, 71], arm gestures [62] and some exercises to strengthen shoulders and arms [67].

Movement classification

The system objective is to classify well and poorly-executed tasks [72–83], to do so many approaches were followed. Some researchers implemented systems to distinguish between normal and abnormal gaits for lower-limb rehabilitation [76, 79, 80], in which participants executed 10 m walks. Other researchers assessed the execution of ADLs [74, 77] like different kitchen-related activities or routine bedroom tasks. Moreover, Lee et.al [72] utilised exercises that belong to popular batteries of tests like Fugi Mayer assessment (FMA) [97, 98], and extension or flexion of elbow and flexibility movements of shoulders [75, 81].

Movement classification englobes as well systems that quantify limb use in order to classify the tasks, in [78], Miller et.al distinguished between uni-manual and bi-manual tasks using both dominant and non-dominant activities while Liu et.al [73] estimated the amount of the affected hand use compared to the unaffected hand. In [82] Derungs extracted digital biomarkers consisting of convergence points physical activity and functional ROM to investigate the affected and less-affected body side. Whereas, Balestra et.al [83], identified different executed tasks in order to count the number of repetitions and determine a correlation with the degree of severity of stroke.

Clinical assessment emulation

In this category, systems that aim to quantify the level of correctness in executing the prescribed exercises are identified. Researchers achieved this by using popular post-stroke assessment scoring systems [84–95]:

FMA variants are the most commonly used batteries of tests from the included works [87, 88, 92, 95], it comprises five domains namely motor functioning, balance, sensation, joint functioning and joint pain in both upper and lower extremities rehabilitation. Scale items are scored on the basis of the ability to complete the item using a 3-point ordinal scale where a score of 0 means the incapacity to perform, 1) a partial performance and 2) a full performance of the task. The total possible score is 226 divided into 100 points for motor functioning, 14 for balance, and 24 for sensation while joint functioning and joint pain have 44 points each. Other variants of this assessment were used such as the short FMA used developed in [98] which includes fewer exercises than the original.

Wolf Motor Function Test (WMFT) [99, 100] is an upper-limb assessment system through timed and functional tasks, the most popular form consists of 17 items in which 6 involve timed functional tasks, 2 are measures of strength, while the remaining consist of analysing the quality of movement quality when performing various activities. It uses a scaling system that ranges from 0 which signifies Does (i.e. no attempt with the limb being tested) to 5 which signifies the attempt was made with a normal-appearing movement. Two included studies used the WMFT [84, 86, 92].

Action Research Arm Test (ARAT) [101] is a 19-item observational measure for upper-limb post-stroke assessment. Items comprising the ARAT are categorised into four subscales namely grasp, grip, pinch and gross movement. Task performance is rated on a 4-point scale, ranging from 0 (no movement) to 3 (movement performed normally).

Two of the included works used the ARAT system [85, 91, 94]

Oxford Grading Motor-Scale (OGM) [102] used in a single study included [89], it evaluates the muscle strength of the rehabilitated patient and can help diagnose problems in which weakness plays a role. It is not proper to stroke rehabilitation and targets both upper and lower extremities. According to the OGM scale, muscle strength is graded from 0 to 5 where 0 implies no muscle contraction and 5 equals movement through a full range against full resistance. Performing OGM requires knowledge of muscle anatomy so that the joints can be positioned correctly as well as the tendon and muscle palpated in order to make a judgement on how much muscle action can be made on the patient.

Chedoke Arm and Hand Activity Inventory (CAHAI) [103], it is an upper-limb post-stroke clinical assessment method that evaluates functional ability. The original CAHAI involved 13 functional items that incorporate a range of movements and grasps that reflect stages of motor recovery following stroke. The clinician will score based on the patient's performance at a scale from 1 which implies a weak performance to 7 which shows complete independence. From the included works Chen et.al used the CAHAI in [90].

The National Institutes of Health Stroke Scale (NIHSS) is a 15-item neurologic scale used to assess the effect of acute cerebral infarction on different levels of consciousness, language, neglect, visual field loss, extraocular movement, motor strength, ataxia, dysarthria, and sensory loss. Scores range from 0 to 42, with higher scores indicating greater severity. A single included paper [83] used this assessment system. Figure 2.1 shows our study taxonomy and the different categories.

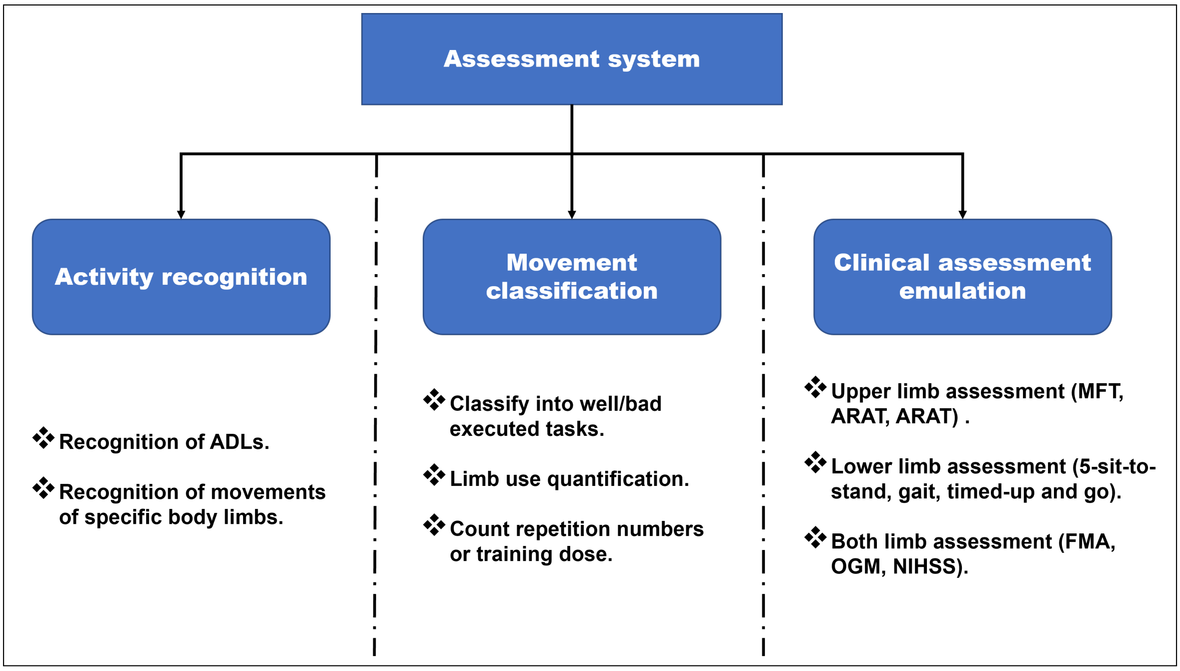

Assessment system
Activity recognition
Movement classification
Clinical assessment emulation
❖ Recognition of ADLs.
❖ Recognition of movements of specific body limbs.
❖ Classify into well/bad executed tasks.
❖ Limb use quantification.
❖ Count repetition numbers or training dose.
❖ Upper limb assessment (MFT, ARAT, ARAT) .
❖ Lower limb assessment (5-sit-to-stand, gait, timed-up and go).
❖ Both limb assessment (FMA, OGM, NIHSS).

Figure 2.1: Proposed taxonomy for post-stroke systems' classification.

2.2.2 Wearable sensors

Over the past few years, effort has been put into developing unobtrusive, effective and objective motion-modeling systems, taking advantage of the progress made in the sensor technology which became more compact and more power-efficient [104]. All the included works utilised IMUs for the data acquisition [61–69, 71–95]. IMUs are devices that combine linear acceleration from accelerometer and the angular turning rates from gyroscopes [105]. IMUs were chosen for their portability and for their low costs, but also because they provide accurate modelling of the participant motion. Some studies used individual accelerometers [62, 64, 73, 74, 84, 86, 87, 89, 90] or gyroscopes [64] while the rest used their combination to give more detailed information. Moreover, IMUs were coupled with different sensors to acquire more information: a barometric pressure sensor to detect changes in altitude [66, 67], insole pressure sensors in [75] to measure the force exercised by the feet while performing the activities, flex sensors to measure the amount of deflection or bending while griping objects [87, 95], liquid level detectors in a cup [61] to measure drinking activity and EMG sensors [70, 81] to measure the activity of the muscles that can translate as strength. Only a single study did not use IMUs and employed EMG sensors only [60].

2.2.3 Sensors' placements

The placement of the sensing technology on the body has shown a heterogeneous distribution linked to the different nature of the employed technology and to the purpose for which the monitoring system was designed. Systems that focused on upper-limb rehabilitation used more frequently the wrists [65, 68, 72, 73, 78, 84, 87, 88, 90–94] in twelve studies, arms [62, 68, 74, 78, 81–83, 86, 88, 89] in ten studies, four studies used forearm [60, 71, 82, 83, 85], three studies used fingers [73, 77, 87], and hands [61, 64, 83], and a single study used elbow and shoulder [87]. These placements were targeted to monitor activities that involved using hands.

By contrast, systems that focused on lower limbs for activities that involved walking utilised more frequently the chest in eight studies [67, 74, 78, 81, 84, 86, 88, 92] the shank in four works [76, 79, 80, 89], thighs [75, 79, 82], and feet [75, 79, 93] in three studies, while two works targeted either the hip [63, 64], or the waist [66, 69], and finally a single study used the lower back [79]. Fig 2.2 shows the targeted placements reviewed in the different studies.

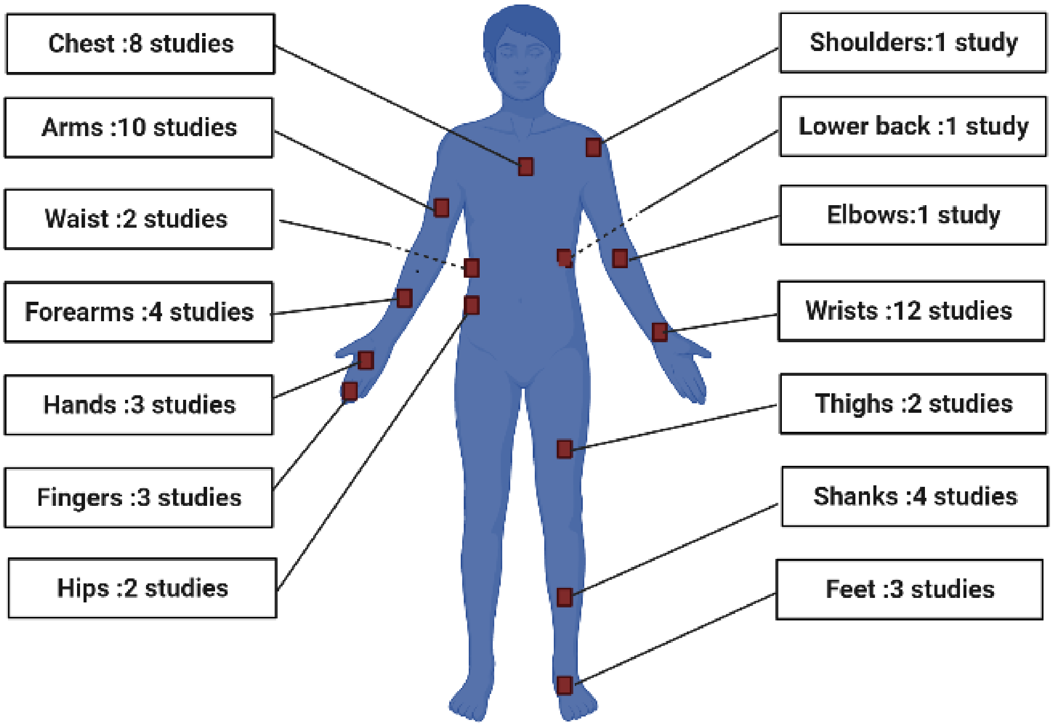


Figure 2.2: Sensors' placements in the included studies.

2.2.4 Study designs and populations

In the included works, different study settings were explored. More commonly it was in controlled environments [106] like labs and hospitals where patients are under the direct supervision of researchers and therapists. Other studies used semi-naturalistic environments [107] where a home environment is replicated in the labs e.g. participants performing their exercises in a kitchen environment under the supervision of researchers. Other studies monitored participants in an outpatient home environment [82].

For the study population, many works recruited stroke survivors with different degrees of severity after getting ethical approvals [62–68, 71, 72, 74, 76–80, 82, 83, 85–94]. Some of them undertook a cohort study by combining them with Able Bodied (AB) participants [62–65, 71, 72, 77, 78, 80, 83, 86] elderly [72] and neurologically disordered patients [79]. Other works only used AB [60, 61, 69], while some studies did not specify or did not use participants [73, 81]. For the number of participants, it varied from 4

SP [89] to 59 SP [90].

2.2.5 Pre-processing and feature engineering

Feature engineering is the process of creating features from raw data to improve the accuracy of a system [108]. Some sophisticated ML algorithms i.e. DL don't require features and can learn to find similarities and differences in raw data automatically [109]. Before selecting features, pre-processing is undertaken on the data to make it ready for the feature study.

According to the different included papers, for filtering unwanted data, designed modules have usually applied threshold-based methods to filter sensor data [62, 63] or used different statistical tools to interpolate the missing data points [83]. Moreover, to filter frequency-based noise, in the frequency domain, other methods are applied such as power spectral density (PSD) [76, 89] Fast Fourier Transforms (FFT) [78, 89], as well as designing different filtering to remove the fluctuations in sensor signals. For example, in [63, 77] noise and unwanted information are filtered out by a low-pass fourth-order Butterworth filter, after that a high-pass fourth-order Butterworth filter was implemented for frequency analysis to eliminate the continuous component of the signal. In [79] Hsu et.al filtered data with a fourth-order bi-directional Butterworth band-pass filter.

Moreover When dealing with accelerometer data, gravity is usually removed from the acceleration as done in [83, 90, 92] by computing the magnitude of acceleration $a(t)$ and subtracting 1. $a(t) = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)}$, in which a_x,a_y,a_z is the acceleration along the x, y, z . The gravity effect can be removed by $VM = |a - 1|$. Compared with raw acceleration triaxial data, VM is insensitive to the gravity effect. In addition, using multiple data sources and thus different sampling rates requires data to be synchronised, to have the same time basis, this has been done by first identifying segments from timestamps and then using linear interpolation as in [83] or padding with zero [81] on the lower frequency data source.

Since the data collected from WS is TS, it should be structured in order to be studied. TS segmentation can be considered either as a pre-processing step for a variety of data mining tasks or as a trend analysis technique. It is also considered as a discretisation problem [110]. A fixed length window is used to segment a TS into sub-sequences and the TS is then represented by the primitive shape patterns that are formed [111]. Segmentation was used by all the papers included herein with time windows varying from 2 s to 10 s depending on the monitored activities. For DL algorithms data after these steps is ready to be fed [64, 74, 80, 84].

By contrast, conventional ML algorithms require further data processing and features that most describe the activities are selected and extracted. In most of the included papers, feature engineering is hand-crafted based on the authors' knowledge of human movements. Time-domain-based features [112] was the most commonly used approach, numerous works extracted Root Mean Square (RMS), mean of time windows, variances, correlatiion between different axes and features, minima and maxima, skewness and other related features [60, 63, 65, 66, 68, 71–73, 75, 77, 78, 81, 85–89, 92, 93, 95]. Some studies coupled it with frequency domain features by converting the data segments using Discrete Cosine Transform (DCT) [61] and extract energy and frequency related features or using Fourier transform [78, 89] and extract frequency components. In [90], Lee et.al used the Discrete Wavelet Transform (DWT) representation to extract wavelet coefficients (Coeffs) and then computed their normalised sum of absolute value. Whereas, in [92] zero-crossing decomposition is applied to the gravity-free acceleration data, to then extract relevant features.

On the other hand, Boukhennoufa et.al in [69] encoded TS windows into GMAF [113] images and fed them into some popular CV algorithms. Studies involving post-stroke rehabilitation require usually many sensors with multiple axes, this could yield huge numbers of features and cause systems to over-fit. To remedy to this, dimensional reduction techniques were used. Dimensionality reduction refers to techniques for reducing the number of input variables in training data by projecting the data to a lower dimensional subspace which captures the essence of the data. Multiple techniques were used in the studies included here. Yang et.al and Tran et.al [60, 65] used a technique called Principal Component Analysis (PCA) that transforms data into fewer dimensions. keeping the three first components only allowed Yang et.al to keep 95.86% of the overall information stored in 56 feature vectors while it allowed Tran to keep 99% of the information, reducing it from the 11 feature vectors. Other studies [63, 78] employed Relief-F which takes a filter-method approach to feature selection to keep only the most relevant features.

2.2.6 Machine learning

ML is an application of AI that provides systems with the ability to automatically learn and improve from experience without being explicitly programmed to do so using the features selected before. Depending on whether to incorporate the outcomes, ML algorithms can be divided into two major categories: unsupervised learning and supervised learning. Unsupervised learning is well known for feature extraction, while supervised learning is suitable for predictive modelling through building some relationships between the patient traits and the outcome of interest [114]. All the papers included used supervised ML algorithms.

Support Vector Machines (SVM) were the most used classifier [61, 63, 65, 68, 73, 76, 77, 87, 89, 93, 95, 115], it was used mainly for classification problems in activity recognition but also in regression problems for clinical assessments where participants are given a clinical score [84–90, 92, 93]. The reasons for choosing SVM variants are their good generalisation ability for sequential data structures [116] and datasets that are not too large. This has been the case in most of the reviewed papers as recruiting post-stroke patients is not an easy task. Moreover, SVM has different kernel types allowing it to deal both with linear and non-linear problems.

Random Forrest (RF) and more specifically Random Trees (DT) were also massively employed [63, 65, 66, 68, 75, 80, 81, 84, 86, 88]. DT is one of the commonest oldest ML algorithm, it models it decision logics to outcomes in a tree-like architecture. Its easiness of interpretation as well as its rapidity of learning made it popular to use in the tele-rehabilitation domain and especially in multi-class activity recognition problems. The reason for that is when going through the tree for a classification sample, the outcomes of all tests at each node will provide relevant information to infer about its class. RF was less used than the former [65, 66, 72, 78], the reason is it is an ensemble of RT making it more prone to over-fitting. It is only used when the available dataset is relatively large.

Artificial Neural networks (ANNs) were also a common choice among researchers for post-stroke rehabilitation assessment. ANNs are a set of ML algorithms that are inspired by the neurons of the brain. ANN may be represented as an interconnection of layers of nodes in which the output of one node is an input to another node for the subsequent processing layer. Multilayer perceptron (MLP) was the commonest among them [61, 75, 77, 79, 80, 115]. MLP does not require feature engineering thus necessitating less domain expertise, although a drawback is the fact that they are considered to be black-box having sometimes unpredictable behaviour. MLP achieved very good results for activity recognition and movement classification. Another ANN architecture that was employed is CNN Architectures [62, 64, 69, 74]. They are designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers [117]. CNNs achieved outstanding accuracies in the CV field but this did not translate to TS data structure which is the structure of the data from the sensors. Boukhennoufa et.al [69], encoded the sequential data into images and then employed a popular CNN architecture which is the Visual Geometry

Group 16 (VGG-16) to achieve very high accuracy.

k-Nearest Neighbour (kNN) is another algorithm that was used in three included works [65, 70, 83]. The kNN classifier is based on distance metrics and was widely used in real-time applications as it is free from the underlying assumptions about the distribution of the dataset. Moreover, The setting of different values for 'K' can result in different classification results for the same problem which makes it an additional hyper-parameter to find the most performing model, especially in activity recognition.

As per the metric to assess the system, almost all the papers used accuracy. It is the proportion of the total number of correct predictions. The accuracy was computed using the formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{2.1}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

Two works [84, 92] used the Coeff of determination, denoted R^2 which is a statistic that will give some information about the goodness of fit of a model in regression models. It was used for the clinical assessment algorithms to compare the predicted score with the score from the clinician.

$$R^2 = 1 - \frac{RSS}{TSS} \tag{2.2}$$

where RSS is the sum of squares of residuals and TSS is the total sum of squares

Moreover, another metric used in regression problems and especially in the clinical emulation assessment [73, 86, 88, 90] is the Root Mean Square Error(RMSE) it is defined to be the standard deviation (SD) of the residuals (prediction errors):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - f_i)^2} \tag{2.3}$$

where d_i is the predicted score value and f_i is the actual one given by the therapist.

2.2.7 Quantitative analysis

In this subsection, a quantitative analysis related to the specifics of the included papers is presented, and statistics about number of citations, system aim, year and 1st author's country of the different works are given.

The system's aim is almost homogeneously divided between the three categories with eleven of the included works treating activity recognition [60–71], twelve dealing with movement classification [72–83] and ten with clinical assessment emulation. [84–95]. This demonstrates that the different categories are of equal interests to researchers.

For the publication year, a growing interest has been noticed starting from 2020 in the included sample of works.

The academic citations of the included papers, ranged from no citations at all [69, 78, 80, 83] to 126 citations [76] (up to August 2021). The papers with no citations were tolerated only in the most recent works written in 2021, and the authors felt it presented an interesting approach worth reviewing. In the same context, statistics of the 1st author's publication are also given. The US, is the country with most publications with thirteen papers [65, 66, 68, 72, 74, 78, 83–86, 88, 89, 92] followed by china with four papers [60, 70, 73, 87], three papers for South Korea [64, 81, 93] and Italy [75–77], two papers for the UK [69, 90] and Thailand [79, 80] and finally a single study for France [61], India [62], Canada [63], Switzerland [67], Germany [82], Singapore [91].

Additionally, as for the targeted limbs from the included papers, upper extremity rehabilitation is the dominant practice with fifteen papers [60, 62, 73, 74, 77, 78, 81, 83, 84, 86–88, 90–92]. This is justified by the fact that most of the clinical assessment batteries of test that the researchers tried to emulate are for the upper extremities, as well as the fact that most of the ADLs involve using hands. Both-limb rehabilitation comes second with twelve papers [61, 64, 65, 67–70, 72, 82, 85, 89, 93] and finally 6 papers for lower-limbs only [63, 66, 75, 76, 79, 80].

2.3 Limitations and challenges

In subsection 2.3.1 the objective and the limitation for each of the included studies are presented, based on that a list of challenges that the researchers in this field most commonly found are extracted in subsection 2.3.1.

2.3.1 Limitations

Challenges

Based on the limitations presented in table A.2, different challenges encountered by the researchers in post-stroke telerehabilitation were assessed.

2.3.2 Quantity and quality of data

ML-based system as for post-stroke smart telerehabilitation, requires rigorous computational models to achieve the desired results and estimate properly the needed parameters. The starting point to construct an efficient model is to have a significant amount of data, besides, the most sophisticated algorithms (e.g. DL) require at least 10 times the number of samples as parameters in the network. Indeed, These algorithms thrive in domains where large amounts of data are easily collected (e.g. CV). On the other hand, in the healthcare area and more specifically in post-stroke rehabilitation, the number of patients is limited, and are not always keen to take part in research projects as it is an extra burden they endure. moreover, more than 70 % of the patients live in low and middle-income countries [118] that do not give enough importance yet to data collection or do not have the necessary means. Subsequently, the available information is still limited to building and training efficient models that would generalise under different conditions and for different cases. Besides, in contrast with other fields where the data is well-structured, healthcare data, in particular and sensor data in general, is heterogeneous, abstruse, noisy and difficult to interpret if not an expert. This makes building a good learning model tricky and requires addressing several challenges, such as data sparsity, missing and dismissed values, sensor miss-calibration issues and noisy segments. In the same context, data bias which is another issue can cause the assessment algorithm to evolve in an unpredictable manner and not generalise to new patients that have different degrees of severity. This was very common among the reviewed papers, where researchers complained about their algorithms not generalising well. [60, 68, 72, 78, 80]. Another data-related issue is confidentiality, especially with the growing use of cloud platforms and the Internet of Things. Therefore, effort should be spent to secure the data transmission between the platforms to ensure privacy for the users of the assessment systems.

Recruitment related challenges

Dealing with post-stroke participants is a sensitive task and requires researchers to follow strict procedures, starting from the recruitment process which requires undertaking tedious ethical approval applications to mounting sensors and collecting data from the participants. In addition, in order to design efficient ML-based assessment systems that generalise to new users a large number of participants should take part with different and variant degrees of severity [62, 67, 68, 88, 89], which is not always available and taken into account. Besides, a common issue found in the included papers when doing cohort studies is not recruiting AB participants that age-match the SP recruited [66, 77], this could yield introduce inequalities that are not caused by stroke disease rather it is by the age difference.

Field complexity and field standards

Understanding illnesses in general and stroke in particular is a more challenging task than dealing with natural language or image processing. It also requires advanced expertise since the systems will be deployed to deal with human subjects to assist them in their rehabilitation process or to assess their execution. Moreover, the standards applied in healthcare are highly rigorous, ethical committees have to approve studies involving human subjects, in addition to the privacy restrictions that govern personal patients' data and sensitive information that limit the use of some modern platforms like computing and data clouds. Furthermore, threats introduced by hacking have become a leading cause of breaches in patients' data, and sensing devices are no exception to this since the information is often transmitted wirelessly. All of these reasons resulted in IoT systems locally processing data [60, 75, 76]. In the same context, some stroke clinical assessments, and some severe cases require particular expertise in dealing with patients to position their limbs, this is usually done with the assistance of experts in the field and is hard to translate to only WS.

Power consumption and latency issues

The WS are continuously sensing data, pre-processing and transmitting it to a remote platform for analysis or visualisation. This results in huge power consumption that may result in the devices turning off and thus terminating the monitoring process of the patients. In addition to that, absolute dependence on cloud platforms for the analysis of data may result in latency of the processing of information due to the huge amounts of data that these platforms receive at once, this may lead to the loss of the real-time aspect of the system or in worst cases to the complete failure of the system when the internet connection is lost.

Patients' acceptance

Patients' approval should be considered in order to build up platforms that will be used in both clinical and home settings. Sensing devices may turn out to be redundant if the patients or clinicians do not use them. Therefore, the wearable device should be unobtrusive, and easy to operate. It should not influence the ADL of the user. Researchers should also concentrate on the implications of the patients' preferences when designing the systems and more efforts should be spent on making stroke patients more familiar with intelligent sensing devices.

2.4 Conclusion and study limitations

The primary contribution of this chapter is two-fold: Firstly, a new taxonomy for post-stroke telerehabilitation assessment was proposed, categorising the field into activity recognition, movement classification, and clinical assessment emulation. The evaluation encompassed various research works conducted in this domain, analysing the utilisation of WSs for data collection. IMU sensors were found to be the most commonly used, with a limited presence of EMG sensors. Additionally, the study examined sensor placements, study designs, feature engineering, and ML techniques employed for the assessment. Secondly, The review identified challenges encountered in the field, including data-related issues, recruitment difficulties, field complexity, power consumption, and patients' acceptance.

In the subsequent chapters, this study aims to address the identified constraints related to field complexity and the adequacy of data volume and quality. The third and fourth chapters focus on enhancing the effectiveness of the evaluation algorithm, while the fifth chapter aims to generate more authentic and informative data. The datasets analysed in this research encompass all forms of assessment systems within the proposed taxonomy, as introduced in the outcome measures Section 2.2.

Chapter 3

Time series models

3.1 Background

As seen in Chapter 2, Human activity recognition (HAR) offers an interesting solution towards the monitoring of ADLs. HAR is a broad field of study that aims to identify the specific movement of a person based on information collected from sensors [119]. The sensor data may be remotely recorded from devices like cameras, radars and force plates or locally recorded using WS. The most challenging task of WS-based HAR in a real-time scenario is to get accurate and reliable information on the patient's activities and behaviours. To do so, many approaches have been investigated, ranging from a conventional signal processing modelling approach that seeks a mathematical relationship between an activity and the different modelling parameters, to ML algorithms, that extract pertinent features to allow the model to differentiate and recognise the different activities [61, 63, 65, 68, 73, 76, 77, 81, 84, 86–89], to more recently DL algorithms that can automatically extract features and learn to distinguish between the activities [61, 62, 64, 69, 74, 75, 115].

In this context, after AlexNet [120] emerged as the winner of the ImageNet competition in 2012, deep CNNs have been successfully applied in various fields [121]. They have achieved remarkable feats such as attaining human-like performance in image recognition tasks [122] and performing diverse natural language processing tasks [123, 124]. Inspired by their accomplishments, researchers have started to adopt these CNN architectures for TS analysis [125]. These are commonly referred to as 1D CNNs. 1D CNNs are better suited to handle 1D signals for the 1D CNNs refer to CNNs with 1D kernel filters while 2D CNNs refer to CNNS with 2D kernel filters. following reasons:

1. 1D convolutions have a lower computational complexity compared to 2D convolutions. This means that a 1D CNN has a lower computational complexity than a 2D CNN under equivalent conditions, like having the same configuration, network, and hyperparameters. 1D models refer to CNN models using 1D kernels while 2D models refer to CNN models using 2D kernels.

2. Most 1D CNN applications have used compact configurations with fewer hidden layers and fewer neurons, whereas almost all 2D CNN applications have used deep architectures with many parameters. Networks with shallow architectures are easier to train and implement.

3. Training deep 2D CNNs requires special hardware setups like cloud computing or GPU farms. In contrast, any CPU implementation over a standard computer is feasible and relatively fast for training compact 1D CNNs with a few hidden layers and neurons.

4. Due to their low computational requirements, compact 1D CNNs are well-suited for real-time and low-cost applications, especially on mobile or handheld devices.

Recent studies have shown that compact 1D CNNs perform better than 2D CNNs for applications with limited labelled data and high signal variations acquired from different sources. Two distinct layer types are proposed in 1D CNNs: CNN layers and MLP layers. The configuration of a 1D-CNN is formed by the number of hidden CNN and MLP layers/neurons, filter (kernel) size in each CNN layer, subsampling factor in each CNN layer, and the choice of pooling and activation functions.

In this chapter, a comprehensive evaluation of thirteen state-of-the-art 1D models is done. These algorithms were originally designed for TS data forecast and were adapted to TS activity recognition. Before feeding the data into these algorithms, the datasets were segmented into multiple time chunks using two different methods:

- **Static segmentation:** The dataset is decomposed into a fixed window, resulting in equal-length time chunks.
- **Dynamic segmentation:** The dataset is decomposed in time chunks, and the segmentation is governed by an event-triggered process.

Two different datasets were utilised: the WISDM Smartphone and Smartwatch Activity and Biometrics Dataset [126] that comprises 19 different activities and of overground and the dataset of treadmill walking kinematics.

3.2 Static segmentation and WISDM dataset:

3.2.1 TS models dataset description and preparation

In the first part of this chapter, the WISDM Smartphone and Smartwatch Activity and Biometrics Dataset [126] was utilised. This dataset was actualised in late 2019. It includes diverse and complex ADL and this makes it a good candidate for evaluating the algorithms. It consists of 18 activities (Table 3.1) performed by 51 different participants for three minutes. Two IMU sensors (triaxial accelerometer and triaxial gyroscope) from a smartwatch and a smartphone were utilised respectively to collect the data. The smartwatch was mounted on the participant's dominant hand, and the smartphone was placed on the waist, with each using a frequency of 20 Hz. Hierarchically, the dataset is divided into two folders, phone and watch, each folder is subdivided into two sub-folders accelerometer and gyroscope, each containing 51 files corresponding to the different participants' IDs. Each file contains the following information: subject-ID, activity-code (character between 'A' and 'S' no 'N' that identifies the activity), timestamp, x, y, z sensors' readings (i.e. accelerometer or gyroscope).

Table 3.1 shows the different activities involved and their labels.

Table 3.1: Dataset activities and their labels.

Activity orientation	Activities
Non-hand-oriented activities	Walking (A), Jogging (B), Stairs (C), Sitting (D), Standing (E), Kicking (M)
Hand-oriented activities (Eating)	Eating soup (H), Eating chips (I), Eating pasta (J), Drinking (K), Eating sandwich (L)
Hand-oriented activities (General)	Typing (F) , Playing catch (O) , Dribbling (P), Writing (Q), Clapping (R), Brushing teeth (G) , Folding clothes (S)

3.2.2 Data pre-processing and classification models

The dataset has been segmented into 10 s chunks corresponding to 200 readings using non-overlapping windows as shown in Figure 3.1 this window length has been chosen in order to compare with the original paper results and other works that used this dataset. Every segment of data is labelled with the most recurring corresponding activity.

All analyses were done using Python (version 3.7.0), with packages (Numpy v1.19.5, Pandas v1.1.5, Scipy v1.4.1). All ML models were trained using either Keras (version 2.4.0) or Tsai (version 0.2.2) from Fastai with Google Collab's Tesla V100 GPU, 25 GB RAM.

To find the best-performing algorithm, sixteen state-of-the-art deep learning classifiers are employed. Different architectures are used namely CNN, RNN, a combination of the two (CNN-RNN), and Wavelet-based neural networks, as previously identified.

A brief description of each algorithm is given below. Besides, three different pre-processing were used on the data: feeding raw data, normalised data, and standardised data to the chosen algorithm. In addition, six different data sources were investigated:

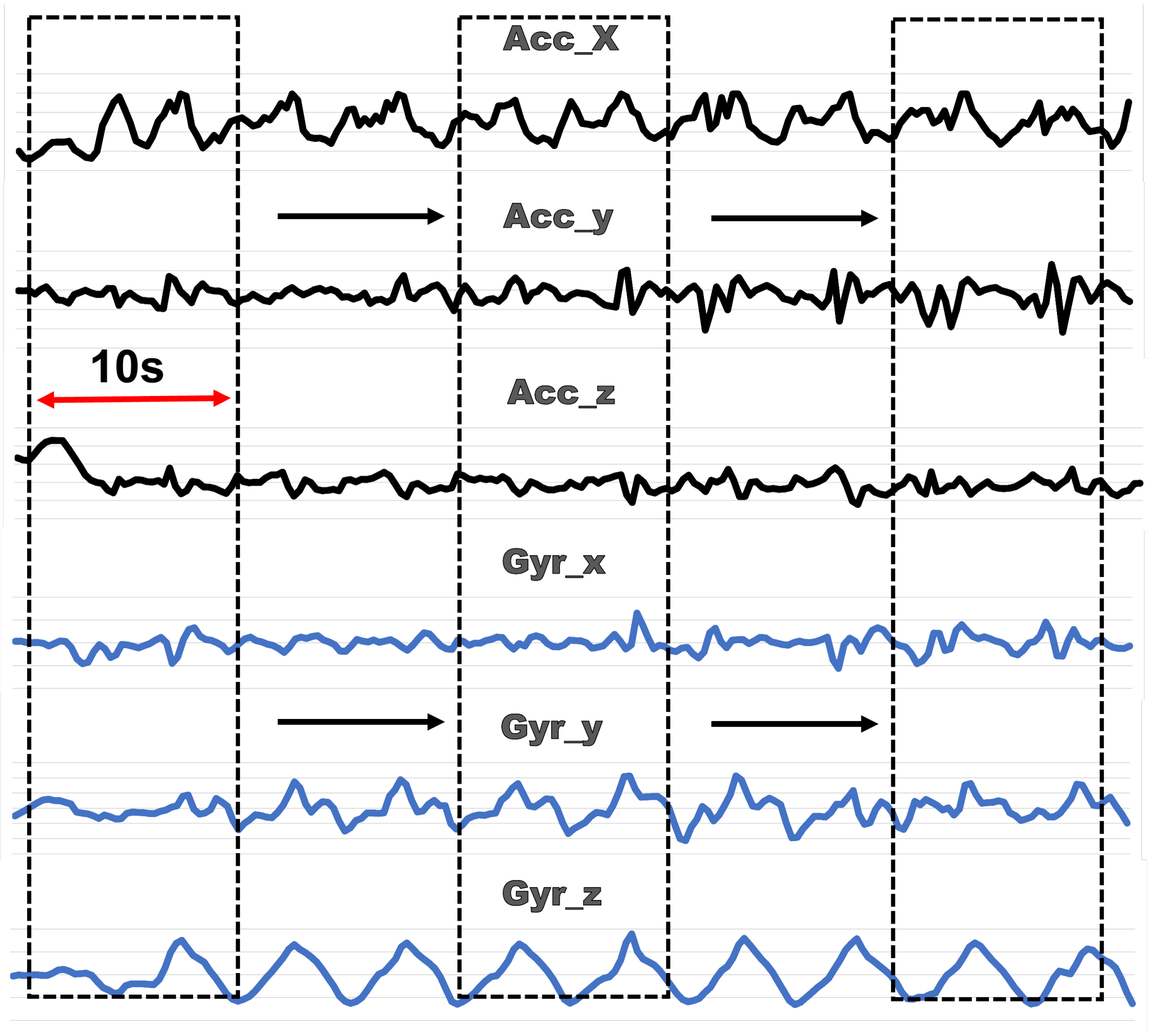


Figure 3.1: Sliding window segmentation.

a gyroscope from the phone, an accelerometer from the phone, a gyroscope from the watch, an accelerometer from watch, a combination of accelerometer and gyroscope from the phone (called phone) and finally, a combination of accelerometer and gyroscope from the watch (called watch). The seventeen different explored models are briefly defined as follows:

- **RNN models**

  RNNs are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. In this paper, six RNNs with different Long

Short-Time Memory (LSTM) are used for HAR, and the main difference between each of them is the number of layers (1, 2, 3) as well as using the bidirectional or non-bidirectional architectures.

- **Fully convolutional Neural networks (FCN):** Inspired by the work introduced by Wang et al [127], it consists of CNNs that do not contain any local pooling layers, meaning that the length of a TS is kept unchanged throughout the layers of convolutions. FCN models have been widely used in various computer vision tasks such as semantic segmentation, instance segmentation, and object detection. They are highly effective in capturing spatial information in images, and they have achieved state-of-the-art performance in many benchmark datasets. It comprises of a series of convolutional layers and pooling layers, followed by a series of upsampling layers that gradually increase the resolution of the output. The final layer of the FCN model uses a softmax function to produce a probability distribution for each pixel, indicating the likelihood that it belongs to a particular class [128].

- **InceptionTime:** Consists of an ensemble of deep CNN models, inspired by the Inception-v4 architecture for computer vision [129]. The composition of an Inception network contains two different residual blocks. For the Inception network, each block is comprised of three Inception modules rather than traditional fully convolutional layers. Each residual block's input is transferred via a shortcut linear connection to be added to the next block's input. Following these residual blocks, a Global Average Pooling layer was employed that averages the output multivariate TS over the whole time dimension. Each inception module contains a bottleneck 1D CNN layer with 32 output channels, a stride of 1 and a kernel size of 1 to reduce parameter dimensionality. The bottleneck layer is followed by three 1D CNN layers with an output channel of 32, a kernel size of 39, 19, 9 consecutively, a padding of 19, 9 and 4, with a stride of 1 in all the cases. The final layer of the InceptionTime network consists of a linear layer to output the internal knee abduction moment.

- **XceptionTime:** is a DL model that is used for TS classification tasks. It is a variant of the Xception model, which is a CNN architecture that was originally designed for image classification tasks. The XceptionTime model extends the Xception architecture to handle one-dimensional TS data [130]. The XceptionTime model consists of a series of depthwise separable convolutional layers, which are designed to efficiently learn spatial features in TS data. These layers are followed by a series of residual blocks, which help the network learn temporal dependencies between the features. The output of the network is fed into a final fully connected layer, which produces the classification output. The XceptionTime model has been shown to achieve state-of-the-art performance on several benchmark TS classification datasets. It is particularly effective when dealing with long sequences of TS data, where traditional recurrent neural network architectures can be computationally expensive and difficult to train.

- **ResNet:** Convolutional layers that stack residual blocks on top of each other to form a network, very popular in the computer vision domain introduced by Kaiming He in [131]. ResNet allows using very deep structures which minimises the problems of vanishing gradients and accuracy saturation, by adding shortcut connections in each residual block to enable the gradient flow directly through the bottom layers [131]. A residual block is a stack of layers set in such a way that the output of a layer is taken and added to another layer deeper in the block. The non-linearity is then applied after adding it together with the output of the corresponding layers in the main path. A TS residual block is comprised of stacking three 1D CNN layers followed by a batch normalisation layer and a ReLU activation layer. The number of filters for the CNN layers in each residual block is 64 then 128 then 256. The final ResNet stacks three residual blocks followed by a global average pooling layer and finally, a linear activation layer to predict the knee abduction moment impulse.

- **XResNet1d:** A modification of the traditional ResNet architecture suggested by Tong He in [132], adapted for TS data.
- **ResCNN:** ResCNN is a type of neural network that improves the training of deep networks by using residual connections. It combines the input of a convolutional layer with its output using a shortcut connection, allowing the network to bypass one or more convolutional layers and create a residual block. This approach helps the network learn the difference between the input and output, reducing the vanishing gradient problem and improving training. ResCNNs have been successful in a range of computer vision tasks, including image classification, object detection, and semantic segmentation. The ResNet architecture is a popular ResCNN model that has been widely used in many computer vision applications.
- **OmniscaleCNN:** A CNN architecture whose specificity is to concatenate the outputs of several convolution filters whose length is one plus all the prime numbers between two and a quarter of the TS length proposed by Tang et al [133].
- **RNN-CNN models**

  A combination of CNN and RNN architectures was investigated, it consists of LSTM layers and convolution layers for feature extraction with different pooling layers.
- **Wavelet-based neural network**

  This model consists of the Multilevel Wavelet Decomposition Network for Interpretable Time Series Analysis (mWDN) Algorithm introduced by Wang et. al in [134]. The particularity of this model is that it preserves the advantage of multilevel discrete wavelet decomposition in frequency learning while enabling the fine-tuning of all parameters under a deep neural network framework.

Experimental results for the TS models

At this stage, raw data were utilised, 75% of the dataset was used for training and the rest for testing, taking a cross-subject split approach to allow the model to learn from all participants. After different training experiments, 32 epochs were found to be the best choice in conjunction with using cyclical learning rates (LR), as proposed by Leslie N. Smith in [135], where the loss is computed and plotted with respect to an increasing LR. The LR is then an interval taken in the range where the loss is decreasing. For example in Figure 3.2, the LR is chosen to be [3e-4,1e-2].

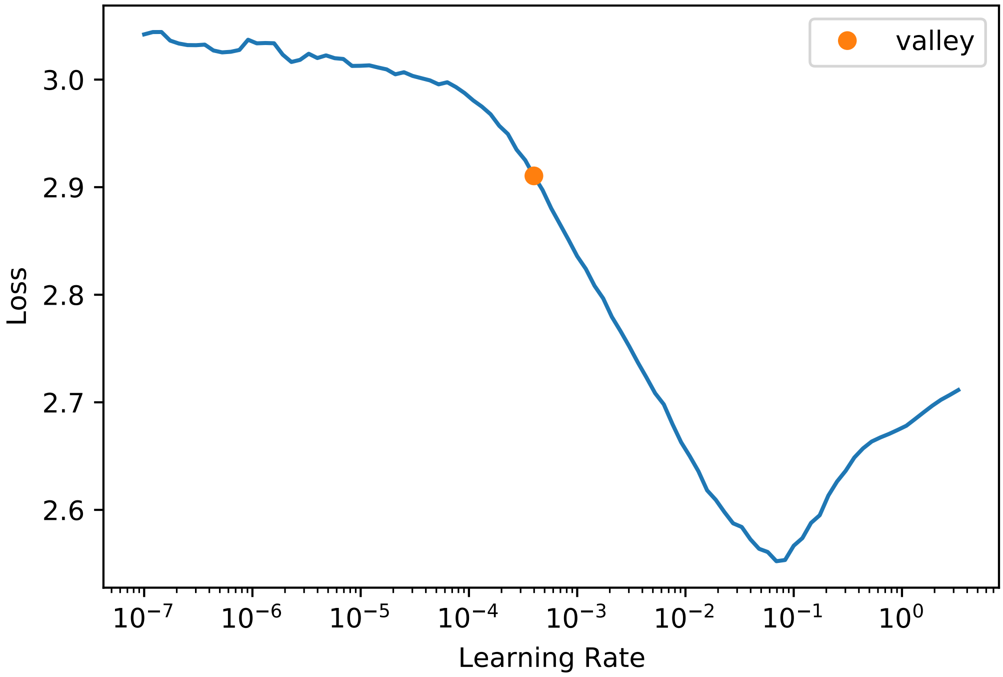


Figure 3.2: Loss by learning rate.

Validation accuracies for the different models with respect to the different sensors's sources are presented then computed. The accuracy was computed using the formula:

$$\text{Accuracy} \ = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

Table 3.2 shows the accuracy results when following the procedure explained earlier, the sensors from the phone perform very poorly on the different models, this can be explained by the fact that the phone was placed in the pocket while some of the activities were hand-oriented, consequently making it difficult for the sensor to sense the changes. The gyroscope from the phone performed best among the sensors, XceptionTime model reached an accuracy of 40.87%.

In contrast, the watch sensors performed significantly better, the XceptionTime gave the best overall results with 77.6% for the accelerometer data and 70.8% for the gyroscope data. While the other models performed slightly worst. This is explained by the fact that the sensor placement this time (wrist) is better for sensing the difference between the hand-oriented activities.

Hence the XceptionTime model has been selected to carry out further fine-tuning and fine-tuning to improve the performance.

Two pre-processing methods were investigated: feeding normalised data and feeding standardised data. Normalisation typically means re-scaling the values into a range of [0, 1] while standardisation means re-scaling data to have a mean of 0 and a SD of 1. The pre-processing was first done on the training data, then the resulting processing parameters were used to per-process the validation data to avoid data leakage.

Results of classification are presented in Table 3.3, only the watch sensors are presented here, the phone data results were discarded because they did not improve from the previous phase.

The combination of gyroscope and accelerometer performed best with 83% and 82%

Table 3.2: Accuracies of the different models on the different sensors' raw data.

Model	Acc watch	Acc phone	Gyro watch	Gyro phone	watch	phone
XceptionTime	0.776013	0.275341	0.707558	0.4087	0.697626	0.299416
ResNet	0.752714	0.224783	0.707558	0.353006	0.704748	0.294947
InceptionTime	0.751655	0.264684	0.683721	0.389305	0.709792	0.275696
ResCNN	0.748478	0.244114	0.680523	0.327237	0.709496	0.27879
LSTM FCN	0.73471	0.225774	0.681105	0.349958	0.707122	0.291165
OmniScaleCNN	0.730209	0.260719	0.680814	0.346911	0.657665	0.263445
LSTMFCN	0.726238	0.245353	0.681105	0.349958	0.697626	0.246476
FCN	0.721472	0.263445	0.677616	0.348573	0.711573	0.289447
xresnet1d34	0.709293	0.253779	0.67907	0.373234	0.680119	0.279821
mWDN	0.67964	0.238662	0.587209	0.313106	0.605242	0.275696
LSTM3	0.666667	0.203965	0.624419	0.365475	0.669436	0.258508
LSTM3bi	0.661371	0.203965	0.612791	0.353006	0.65905	0.237882
LSTM2	0.657665	0.199009	0.604942	0.343585	0.623739	0.232726
LSTM2bi	0.656606	0.17596	0.603198	0.337213	0.619881	0.217944
LSTM1	0.605242	0.174226	0.573837	0.326129	0.60178	0.202475
LSTM1bi	0.587503	0.163569	0.556395	0.321142	0.588131	0.196631

for the standardised and normalised data successively. The confusion matrix for the best performing one (standardised) is shown in Figure 3.3. The ExceptionTime model achieves near-perfect classifications for non-hand oriented activities (A, B, C, D, E, M) and the general hand activities (F, G, O, P, Q, R, S) whereas it has more difficulty differentiating between the different eating activities (H, I, J, K, L) especially eating sandwich (L) eating chips (I) and eating pasta (J).

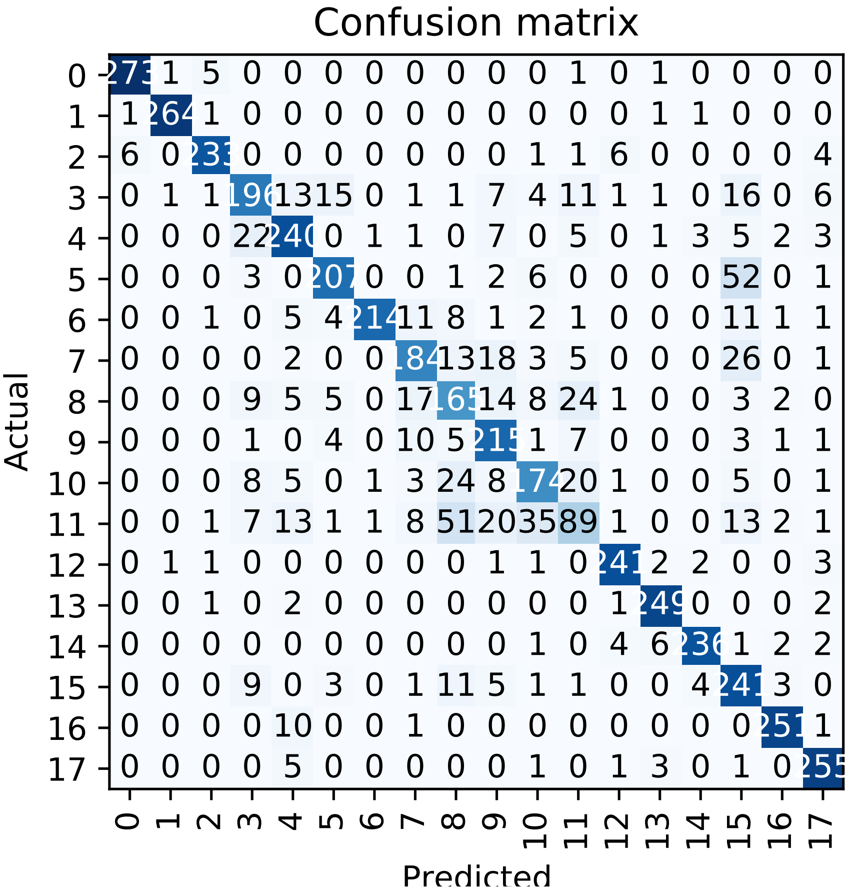
Table 3.3: Validation Accuracies of the different watch sensors' configurations

Sensor	Standardised	Normalised
Accelerometer	81%	69%
Gyroscope	73%	74%
Combination	83%	82%

3.3 Dynamic segmentation and treadmill walking kinematics dataset:

This was a secondary analysis of a publicly available dataset, using a single session, cross-sectional laboratory study design [136]. The data came from a public dataset of 42 healthy adults walking on a treadmill, the details of which can be found in the original open-source publication [136]. Nine out of the 42 participants from the walking dataset were excluded from the present study. These participants had simultaneous bilateral foot contacts on the same force plate, resulting in an absence of consecutive good foot contact strides which lasted >50% of the walking duration. The 50% threshold was determined by the authors to minimise manual identification of foot contact events, and to increase processing replicability [137].

Participants performed unshod walking on a dual-belt, force-instrumented treadmill (300 Hz, FIT; Bertec, Columbus, OH, USA), and motion was captured with 12 optoelectronic cameras (150Hz, Raptor-4; Motion Analysis Corporation, Santa Rosa, CA, USA) [136]. This dataset was deemed feasible for this study given that the primary aim is to determine the optimal network architecture for using TS kinematic measures to predict knee joint moment impulse. Walking occurred over eight controlled speeds: 40%, 55%, 70%, 85%, 100%, 115%, 130%, and 145% of each participant's self-determined dimensionless speed (Froude number). The associated absolute walking



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	273	1	5	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
1	1	264	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
2	6	0	233	0	0	0	0	0	0	0	1	1	6	0	0	0	0	4
3	0	1	1	196	13	15	0	1	1	7	4	11	1	1	0	16	0	6
4	0	0	0	22	240	0	1	1	0	7	0	5	0	1	3	5	2	3
5	0	0	0	3	0	207	0	0	1	2	6	0	0	0	0	52	0	1
6	0	0	1	0	5	4	214	11	8	1	2	1	0	0	0	11	1	1
7	0	0	0	0	2	0	0	184	13	18	3	5	0	0	0	26	0	1
8	0	0	0	9	5	5	0	17	165	14	8	24	1	0	0	3	2	0
9	0	0	0	1	0	4	0	10	5	215	1	7	0	0	0	3	1	1
10	0	0	0	8	5	0	1	3	24	8	174	20	1	0	0	5	0	1
11	0	0	1	7	13	1	1	8	51	20	35	89	1	0	0	13	2	1
12	0	1	1	0	0	0	0	0	0	1	1	0	241	2	2	0	0	3
13	0	0	1	0	2	0	0	0	0	0	0	0	1	249	0	0	0	2
14	0	0	0	0	0	0	0	0	0	0	1	0	4	6	236	1	2	2
15	0	0	0	9	0	3	0	1	11	5	1	1	0	0	4	241	3	0
16	0	0	0	0	10	0	0	1	0	0	0	0	0	0	0	0	251	1
17	0	0	0	0	5	0	0	0	0	0	1	0	1	3	0	1	0	255

Figure 3.3: Confusion matrix of the normalised data from the watch.

speeds for all eight conditions for each participant were reported by the authors [136]. Marker trajectories and ground reaction force (GRF) were low passed filtered at a matched frequency of 6Hz (4th Order, zero-lag, Butterworth) [137]. A seven-segment lower limb, the 6DOF joint model was developed in Visual 3D software (C-motion Inc., Germantown, MD, USA) [137]. A force plate threshold of 50N was used to determine gait events of initial contact and toe-off.

Three-dimensional (3D) angular and linear displacement, velocity, and acceleration of the seven-segment's centre of mass (COM), relative to a fixed global coordinate system were derived and formed the predictor space (126 TS predictors). These kinematic predictors were used as they represented predictors that can potentially be measured using IMUs. Internal moments are automatically calculated in Visual 3D. Hence, the internal knee abduction moment (inverse of the external KAM) was calculated using inverse-dynamics and expressed in the proximal segment's reference frame [138] (negative values indicated internal knee abduction moment)

Machine learning modeling

All analyses were done using Python (version 3.7.0), with packages (Numpy v1.19.5, Pandas v1.1.5, Scipy v1.4.1). All ML models were trained using either Keras (version 2.4.0) or Tsai (version 0.2.2) from Fastai with Google Collab's Tesla V100 GPU, and 25 GB RAM.

Generic pre-processing

Dynamical segmentation of all TS (predictors and outcome) was performed by an event-triggered algorithm that takes into account the right foot on the ground: between initial contact (RON) and toe-off (ROFF) [137], as shown in Figure 3.4, resulting in non-fixed time chunk, and this is referred to as dynamic segmentation that differs from the static segmentation utilised in section 3.2.2. For the outcome, the area under the (negative) internal knee abduction moment curve for each TS segment was calculated to provide a measure of knee abduction impulse. The knee abduction impulse was normalised to each participant's body mass (Nm.s/kg). Given that the stance duration between each step, each speed condition, and participants were different, each TS segment had a different number of data points. the TS segments were zero-padded to have an equal number of data points as that of the longest TS segment [139, 140].

Three different pre-processing methods and their influence on prediction performance were explored: (1) using raw TS data as predictors, (2) normalising the TS predictors to a range from 0 to 1, and (3) standardising the TS predictors to a mean of 0 and SD of 1. Although scaling of predictors (e.g. to a mean of 0 and SD 1) is commonly

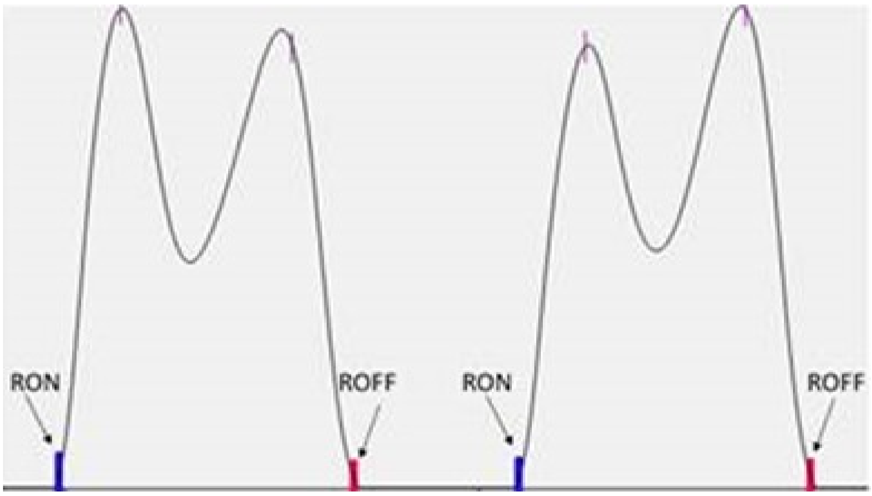


Figure 3.4: Dynamic segmentation

advocated in ML [141], the best prediction performance was found to be provided by using raw TS as predictors in the exploratory analysis, and this was subsequently utilised in formal ML modelling.

The total number of observations in the dataset was 6737 corresponding to 6737 participant-steps. The predictor dataset was organised into a 3D array of shapes 6737×126 ×300, where the second dimension was the number of predictors, and the third dimension was the number of time points. The outcome dataset was organised into a 1D vector of length 6737. Both the predictor and outcome datasets were split into training (75%, n = 5052) and testing (25%, n = 1685) datasets [142]. The training dataset contains 75% of all the participants' data with all the controlled speeds while the test dataset contains the rest of the dataset over the controlled speeds. This allows the model to learn from all the different cases to permit a more robust generalisation for each distinct instance. Our method of ML model development relies on a scenario which a participant comes for a baseline biomechanics assessment to develop a personalised model for the prediction of future instances of knee joint loads.

Figures (3.5-3.11), show a plot of the mean of each predictor around all the segments and in shadow ± the SD:

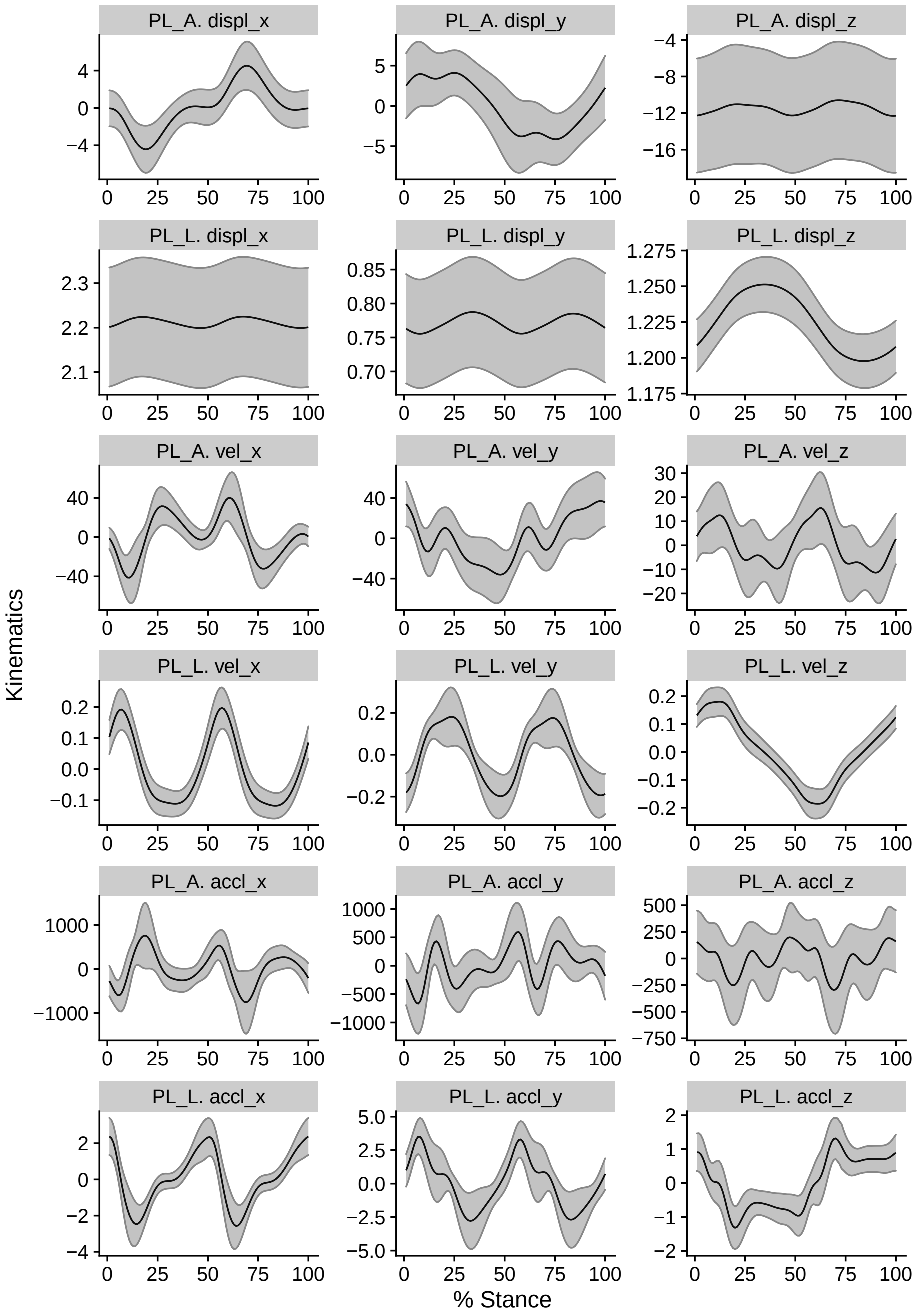


Figure 3.5: Predictors 1-18

Algorithms

The following architectures were evaluated: 1) A 2D CNN-based model used as a baseline model, 2) the InceptionTime model, 3) transfer learning, and 4) the TS-Resnet

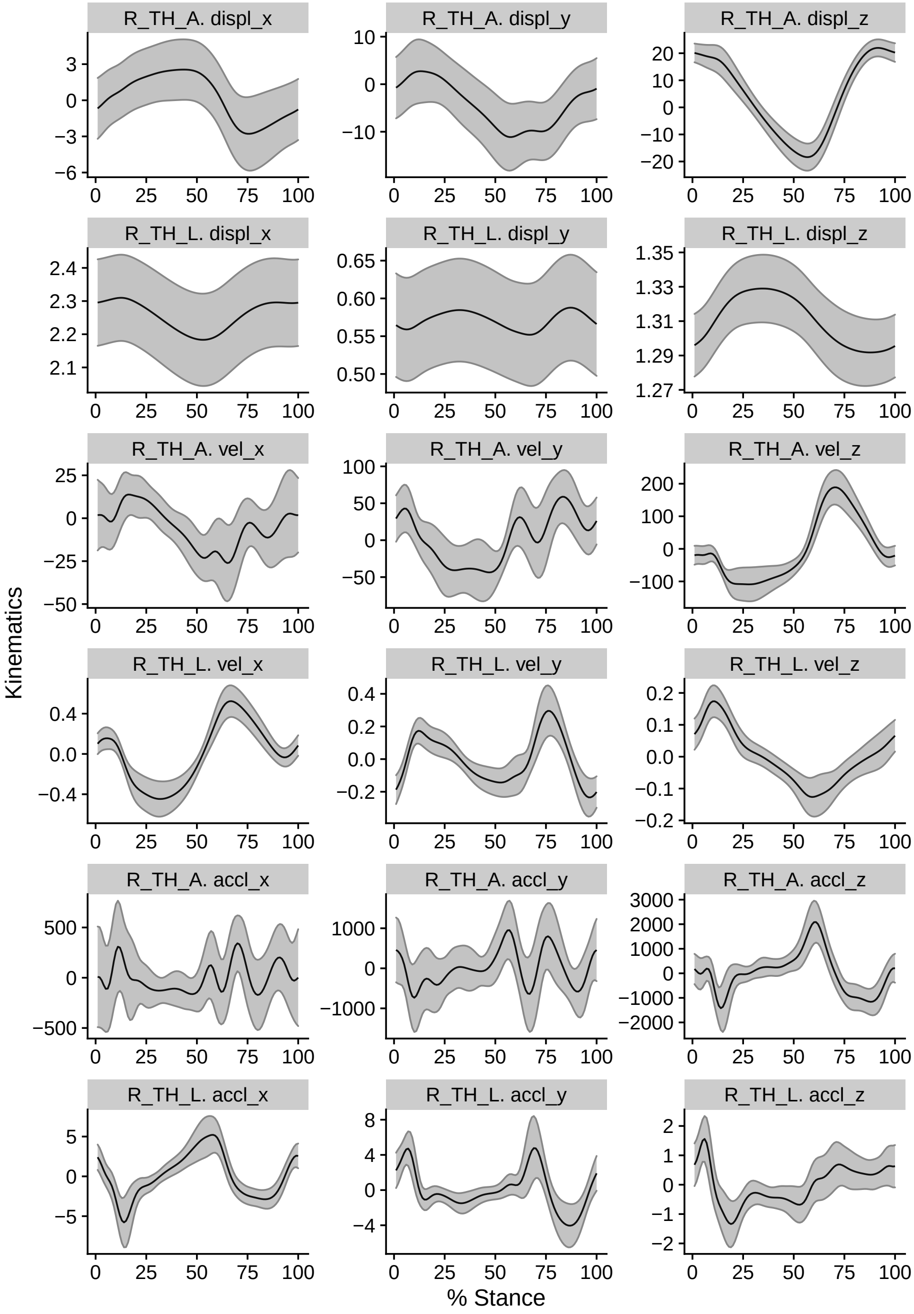


Figure 3.6: Predictors 19-36

model.

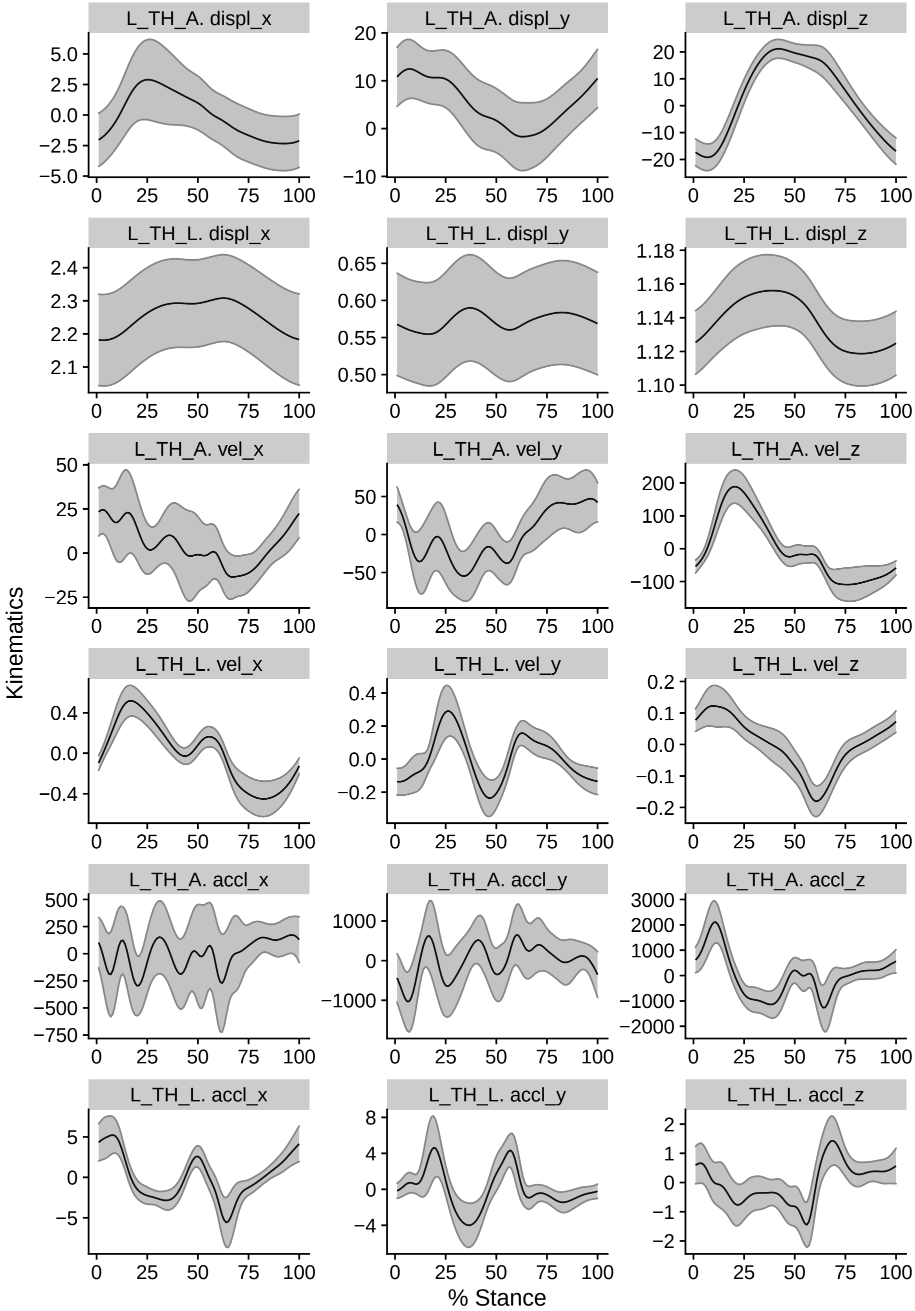


Figure 3.7: Predictors 37-54

2D CNN model

The baseline 2D CNN model architecture can be found in Figure 3.12. Convolutional layers in a neural network are designed to learn a hierarchical representation of local

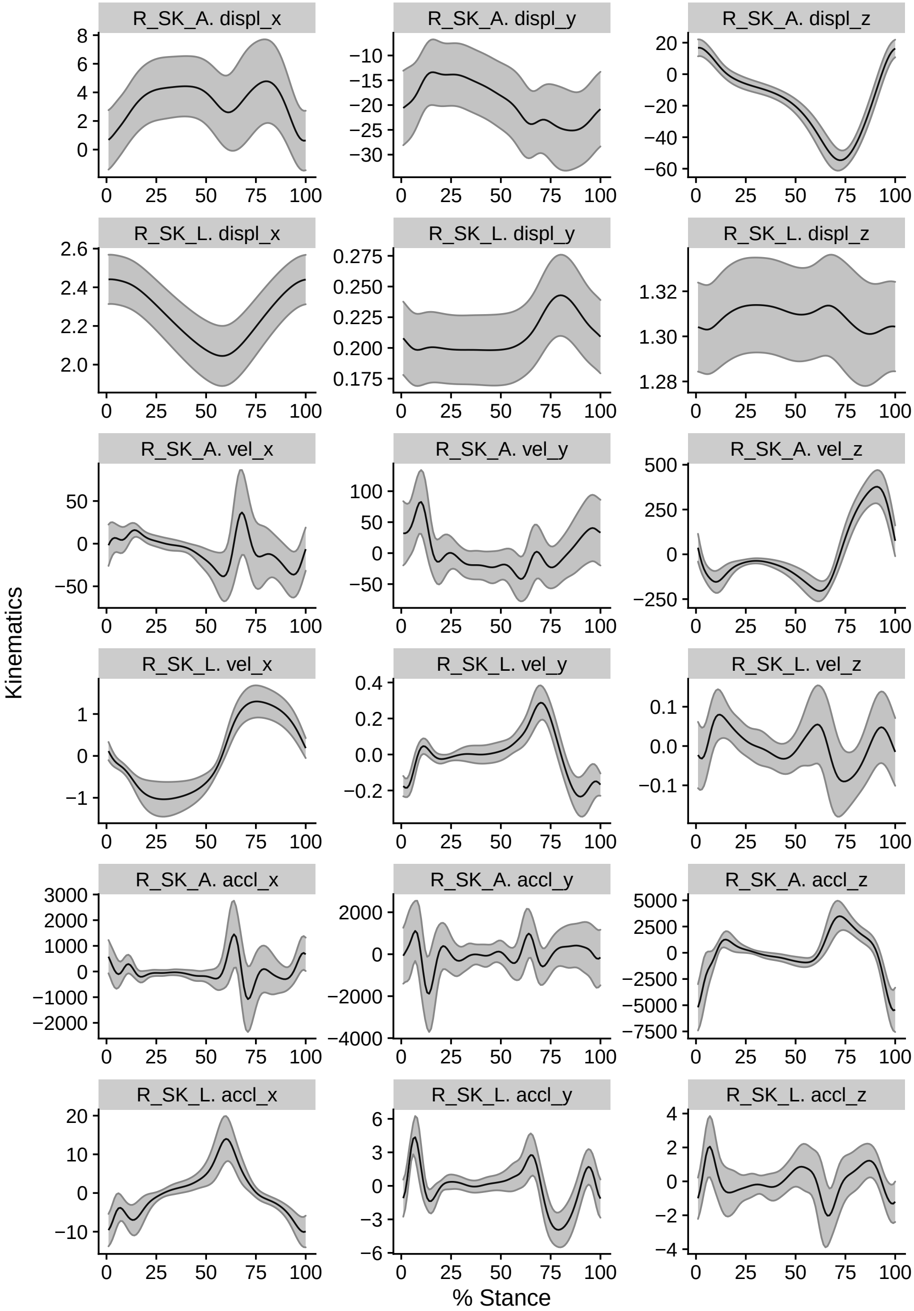


Figure 3.8: Predictors 55-72

features (e.g. peaks) of the predictors [143]. Advantages of convolutional layers over fully connected layers include having to learn much fewer parameters, better generalisability, and better scalability to big datasets. The model hyperparameters

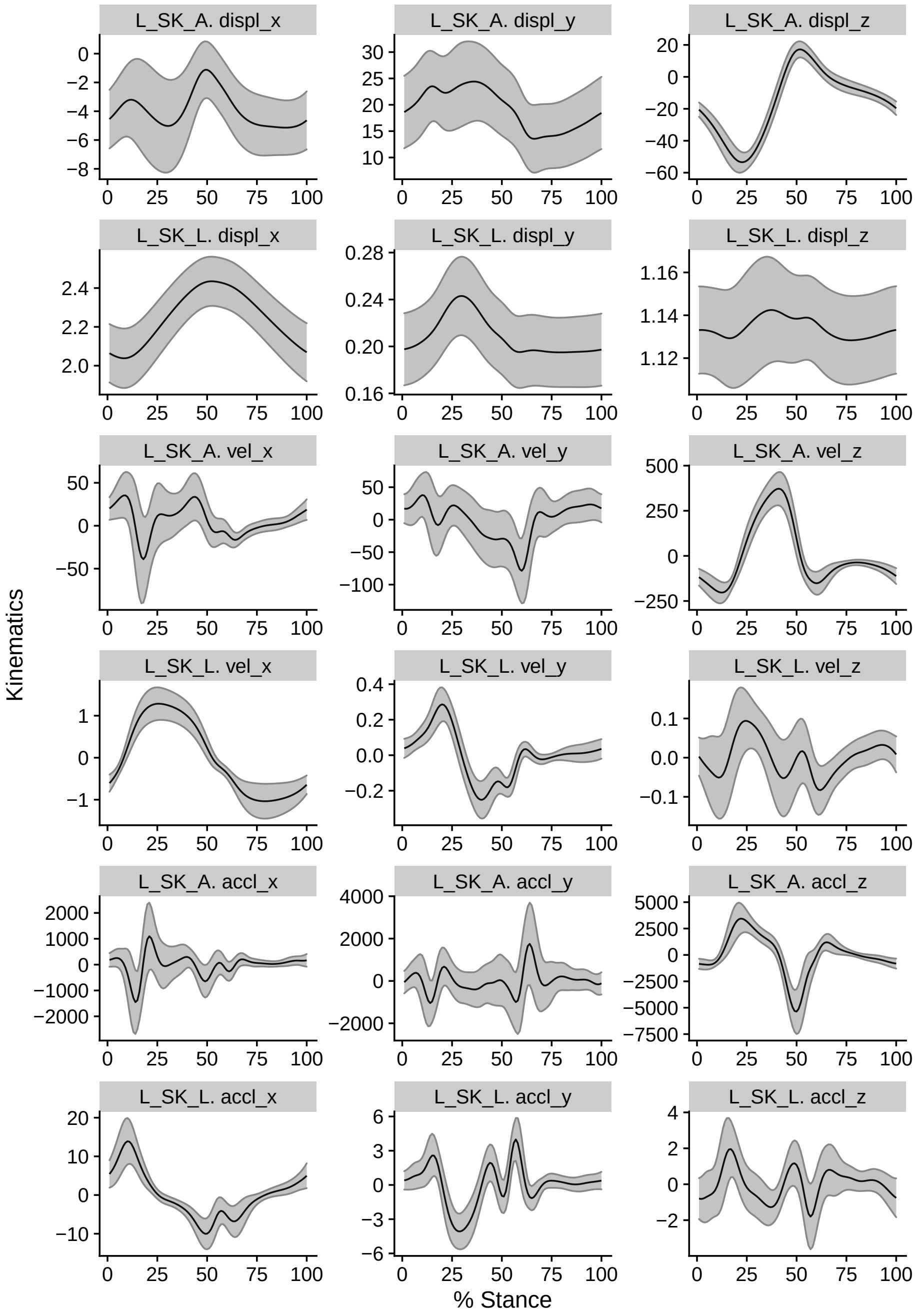


Figure 3.9: Predictors 73-90

were selected based on initial exploratory analysis. Neural network (NN) weights were initialised with Xavier initialisation [144]. The Xavier initialisation method is calculated as a random number with a uniform probability distribution (U) between

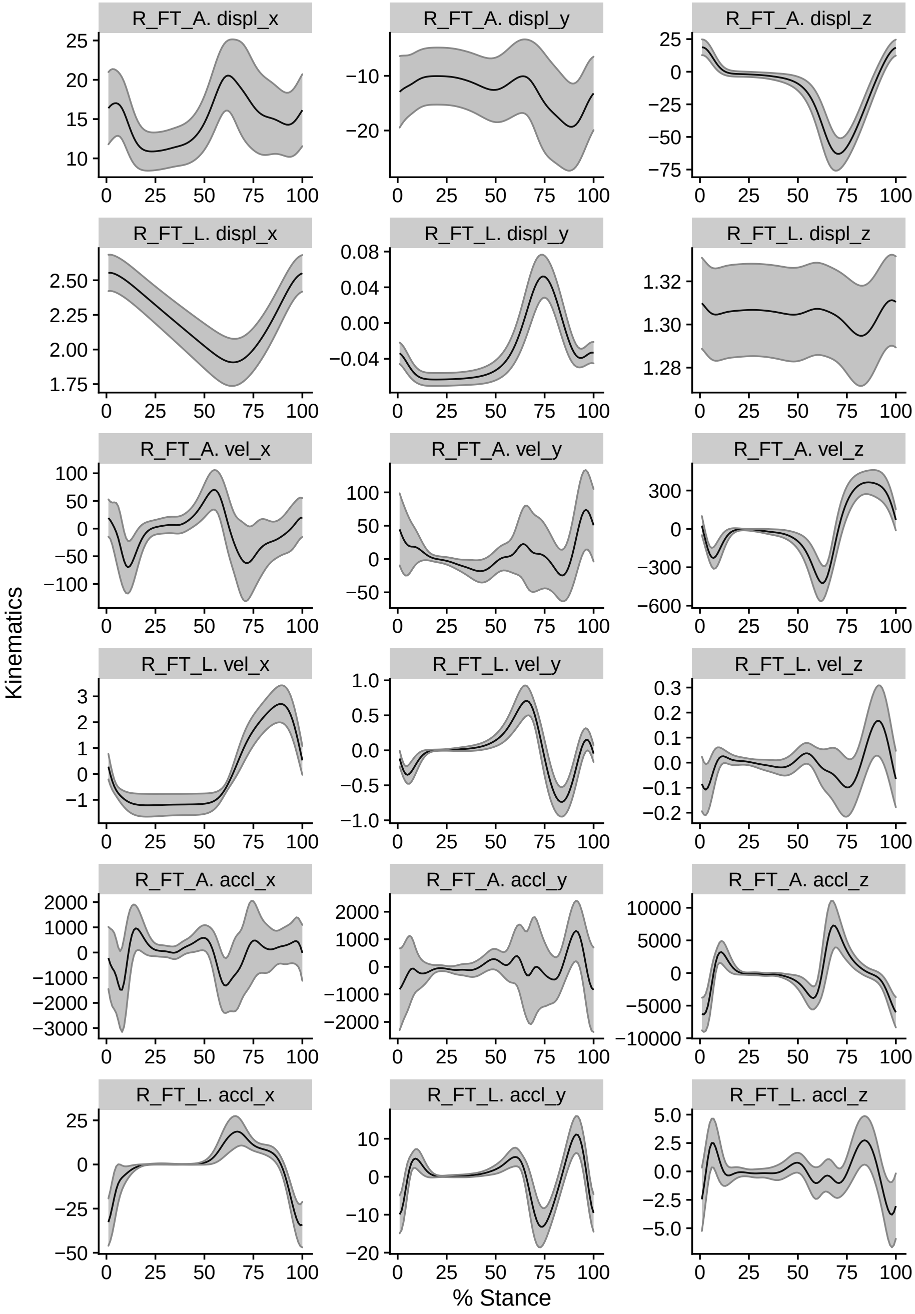


Figure 3.10: Predictors 91-108

the range $-\frac{1}{\sqrt{n}}$ and $\frac{1}{\sqrt{n}}$, where n is the number of inputs to the node.

$$\text{weight } = U\left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right] \tag{3.2}$$

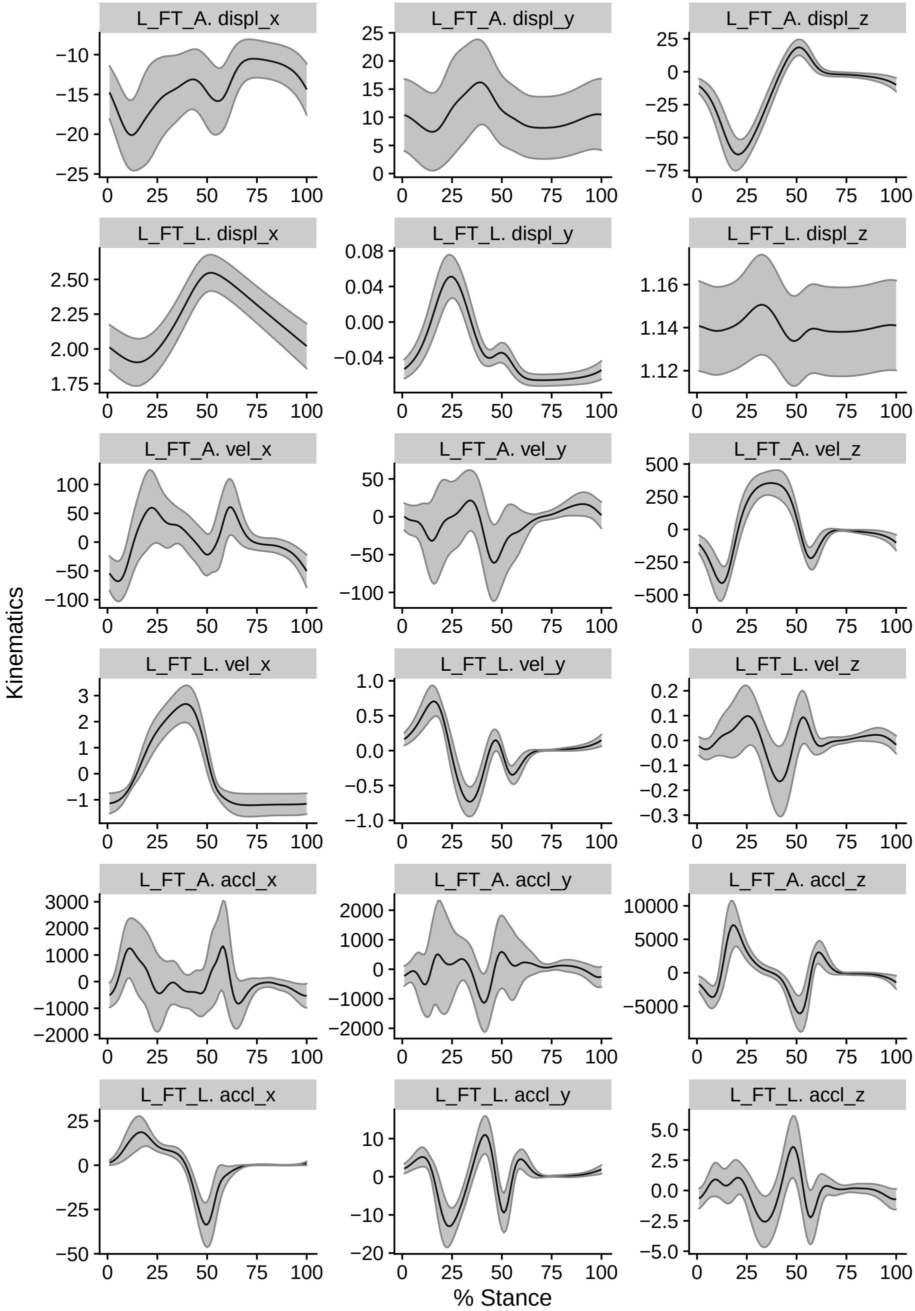


Figure 3.11: Predictors 109-126

A batch size of 64, 100 epochs of training repetitions, an LR of 3e-3, and an Adam optimiser were used. The mean squared error as the loss criteria were also used.

For the other neural network models, a different method to find the appropriate LR,

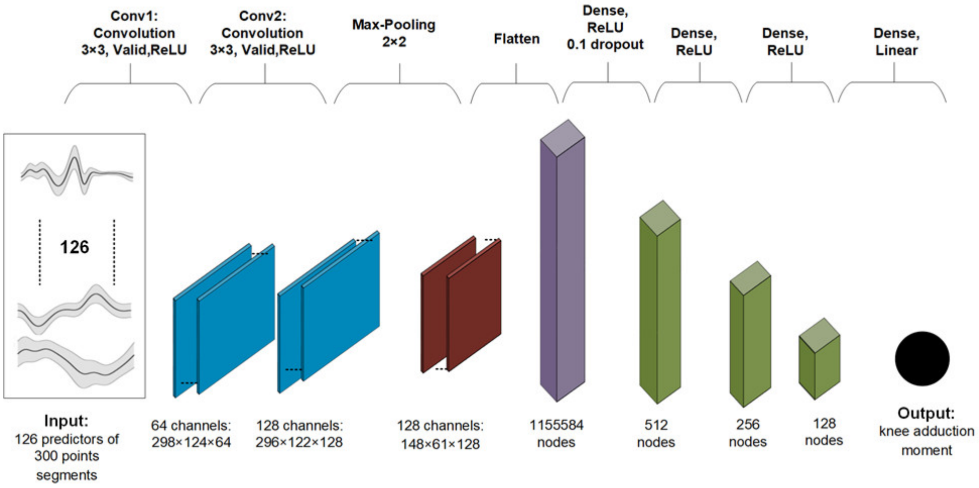


Figure 3.12: Baseline two dimensional convolutional neural network architecture.

which has been termed cyclical LRs, was used [135]. The loss was plotted with respect to an increasing value of the LR. The LR was chosen to be in the interval that resulted in the lowest loss, which was found to be between 8e-3 to 1e-1. The LR took the value of 8e-3 at the first epoch and then gradually increased to reach a final value of 1e-1 at the last epoch. In conjunction with the cyclical method, it was found that after only ten epochs the loss stabilises and therefore 10 epochs were chosen, and a batch size of 128. weights were initialised with Xavier initialisation. The three models used are described below:

InceptionTime

Transfer learning InceptionTime

The InceptionTime model previously defined that was pre-trained on datasets from the UCR archive [145]. Only two layers were tuned from the InceptionTime model - the first input layer to ensure that the required dimensions of the data conformed to our dataset; and the last layer in which the activation function was changed to linear to predict the continuous outcome of knee abduction moment impulse.

TS-Resnet

ResNet allows using very deep structures which minimises the problems of vanishing gradients and accuracy saturation, by adding shortcut connections in each residual block to enable the gradient flow directly through the bottom layers [131]. A residual block is a stack of layers set in such a way that the output of a layer is taken and added to another layer deeper in the block. The non-linearity is then applied after adding it together with the output of the corresponding layers in the main path. A TS residual block is comprised of stacking three 1D CNN layers followed by a batch normalisation layer and a ReLU activation layer. The number of filters for the CNN layers in each residual block is 64 then 128 then 256. The final ResNet stacks three residual blocks followed by a global average pooling layer and finally, a linear activation layer to predict the knee abduction moment impulse.

Predictive performance

The prediction performance of the knee abduction impulse was calculated on our test dataset using the metrics of the RMSE, (Nm.s/kg), the mean average error (MAE, Nm.s/kg), and the mean absolute percentage error (MAPE, %), and the normalised root mean squared Error (NRMSE, percentage). The RMSE was computed using

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}} \tag{3.3}$$

where y is the observed knee abduction moment, y is the predicted moment and N is the number of observations in the test dataset. The MAE was computed using

$$MAE = \frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|}{N} \tag{3.4}$$

The MAPE was computed using:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left\|\frac{y(i)-\hat{y}(i)}{y(i)}\right\| \tag{3.5}$$

And finally, the NRMSE which is the RMSE divided by a measure spread. In this work, the RMSE is divided by the difference between the min and the max of the knee abduction.

$$NRMSE(\%) = \frac{RMSE}{MAX - MIN} \times 100 \tag{3.6}$$

The max and min values are reported below in the result section.

3.3.1 Results on the treadmill dataset

For the 33 included participants (female = 15, male = 18), the mean SD age was 39.42 (17.87) years, height was 1.67m (0.12m), and body mass was 67.66 kg (12.44 kg). The mean knee adduction impulse was -28.06 Nm.s/kg, SD was 11.55 Nm.s/kg, the interquartile range was 15.17 Nm.s/kg, with a variation range (max-min) of 86.94 Nm.s/kg. The mean (SD) waveforms of our 126 predictors normalised 100% timepoints of the stance phase, on our dataset can be found in the supplementary material.

Table 3.4: Regression models performance

	Training set loss(Nm.s/kg)	Validation set loss(Nm.s/kg)	Test set MAE (Nm.s/kg)	Test set RMSE (Nm.s/kg)	Test set MAPE (%)	Test set NRMSE (%)
Baseline model	8.91	16.97	2.78	3.46	10.80	3.98
InceptionTime	6.70	6.05	1.76	2.46	8.61	2.83
Transfer learning	6.54	5.59	1.70	2.36	8.28	2.71
TS-ResNet	5.28	6.10	1.77	2.47	8.65	3.15

Table 3.4 shows the performance of the four ML models. MAPE and MAE are measures of how far the model's predictions are off from observed values on average.

The baseline model achieved 10.80% with the predicted value spreading on average 2.78 Nm.s/kg from the observed values. Transfer learning with inception time was the best performing model, achieving the best MAPE of 8.28%, which translates to the predicted value spreading on average 1.70 Nm.s/kg from the observed values. In contrast, training the inception time model from scratch resulted in a slightly lower performance compared to transfer learning, with a MAPE of 8.61%. The GADF-xResnet 18 model performed worse than the baseline model with a MAPE of 16.17%. This means that converting a TS to images did not improve ML prediction performances.

3.3.2 Discussion about the dynamic segmentation

The ability to quantify joint moments in the field may revolutionise the clinical management of musculoskeletal disorders where tissue loading has been implicated as a risk factor for the onset, exacerbation, and symptomatic relapse. In partial support of our hypothesis, transfer learning resulted in the best prediction performance of the outcome of knee abduction impulse during walking. However, in contrast to our hypothesis, the GADF-xResnet model was the worst-performing algorithm.

The only other study to our knowledge that investigated the accuracy of ML in predicting KAM impulse was Stetter et al. [146], which reported an average observed value of 69.16 Nm.s/kg, and a predicted value of 64.23 Nm.s/kg – a difference of 4.93 Nm.s/kg. Given that performance metrics (RMSE, MAE) were not reported for KAM impulse [146], the difference in average values as the performance metric were used for comparison. The performance in predicting KAM impulse in the previous study [146] was worse than all our models tested in the present study. The worse performance by Stetter et al. [146] could be due to two reasons. First, the previous study used IMU TS predictors [146] which may be noisier than our kinematic predictors. Second, Stetter et al. [146] performed validation whereby the training and testing data were independent (i.e. subject data in the training set not in the testing set). However, our training and testing data were dependent, the reason for which was explained in the methods section. Third, they used a fully connected layered neural network model which may not adequately harness the temporal information within the variables [146]. As previously mentioned, Boswell et al. [147] reported that a fully connected network was superior to an LTSM network, but the poorer performance of the latter could be an insufficient number of layers. Future investigations are needed to benchmark different types of network architectures on different biomechanical datasets to determine when different modelling approaches would be superior.

A number of layers used in our ResNet model was potentially insufficient to learn the parameters.

Another finding of the present study was that our baseline CNN model performed worse than InceptionTime and TS-ResNet, using the same TS predictors. Both InceptionTime and TS-ResNet contain shortcut residual connections between convolutional layers, whilst our baseline CNN model does not. The benefit of having residual connections within the network is that makes training a deep neural network much easier by reducing the vanishing gradient effect [131]. In addition, the high performance of InceptionTime may be attributed to having multiple parallel convolutional layers, each with different filter lengths, learning different latent hierarchical features of the TS. The benefit of having multiple parallel layers may be analogous to the benefit of ensemble ML techniques like boosting – combining the results of multiple weak learners. InceptionTime, when compared to the baseline model, combines multiple extracting structures with different window sizes, which allows the former to extract a more diverse set of features from the predictors than the latter, thereby improving the prediction performance using InceptionTime. In a consistent manner, TS-ResNet's deep architecture also allows to learn a plethora of features that are associated with this dataset. In contrast, the baseline model likely did not allow to learn the features as well as with InceptionTime and TS-ResNet due to its shallow architecture. Interestingly, our finding that transfer learning resulted in the best prediction performance was not supported by another study, albeit conducted in running [148]

The following limitations were identified in this study. First, hyperparameter tuning was not performed, and the selected hyperparameters were chosen based on the authors' experience. Therefore, our findings could be considered to provide a more conservative estimate of the predictive performance of deep and transfer learning models. Second, predictors derived from optoelectronic systems were used, which can still be a time-method to use in clinics. WS or markerless motion capture represents the most clinically feasible methods of measuring body motions. Whether the performance of the ML approach using these newer technologies would match that of traditional optoelectronic systems needs to be investigated. Third, the model was developed using data collected from treadmill walking, and the performance may differ in overground walking. Lastly, the models were trained to predict a specific load metric, the internal knee abduction impulse. Whether the present study's findings would similarly translate to other knee load metrics (e.g. peaks), or indeed the entire TS curve, will require investigation..

3.4 Conclusion

In this chapter, an investigation into the performance of various state-of-the-art TS DL algorithms was conducted. Two different datasets were used: the first was a complex HAR dataset, and the second was aimed at predicting the knee abduction moment impulse. The impact of various factors such as sensor utilisation, sensor placement, pre-processing algorithms, and transfer learning on the algorithm's performance were analysed.

The analysis of the first dataset revealed that the different algorithms produced accuracy rates that were relatively close to each other. However, the Xceptiontime model slightly outperformed the other models. It was observed that sensor placement played a significant role in accurately recognising the activities, as some placements were more sensitive to specific activities than others. In particular, a maximum accuracy rate of 42% was achieved by using the algorithm on data from sensors placed on the waist, while the sensors placed on the hand yielded an accuracy rate of 84%.

On the second dataset, it was found that TS-based DL models were highly effective in predicting knee abduction moment impulse during walking. It was also observed that transfer learning improved the predictive model's performance, even though the two models were derived from different domain disciplines. The results supported the idea that combining ML with kinematic inputs can effectively quantify biomechanical kinetic measures outside the laboratory.

In the previous chapter, the application of CV models to TS data in rehabilitation assessment was examined, revealing promising outcomes and demonstrating the models' adaptability and strength. As the focus shifts in the next chapter, a novel transformative approach called 'Imaging TS' is introduced. This technique transforms time TS into visual images, which are then analysed using 2D CNNs. This approach not only explores new methods but also addresses previous challenges related to algorithm performance and data interpretation. It leverages established CV models for a deeper understanding and analysis of time series data, marking a significant step into new research territory in the field of rehabilitation assessment. A complete pipeline that will also use interpolation to increase/decrease image size is also presented.

Chapter 4

Encoding time series models

4.1 Background

In the preceding chapter, an attempt was made to address the issue of rehabilitation assessment performance, as presented in Chapter 2, by adapting CV models to the TS (TS) format. This involved segmenting data either statically or dynamically, modifying the structure of conventional CV models using one-dimensional (1D) kernels, and rearranging the input to make it compatible with the TS format.

In this chapter, a different approach is followed, whereby the TS data chunks are encoded into images and fed into a two-dimensional (2D) kernel CNN model, which is also a CV model. This approach, known as “imaging TS," is increasingly being utilised. For example, Souza et al. [149] used recurrence plots to encode TS from different univariate UCR datasets. Images were then fed to support vector machine (SVM) classifiers and outperformed state-of-the-art methods at the time. Researchers in [113] employed GMAF and MKV to encode five multi-disciplinary univariate TS datasets. The resulting images were then fed to tiled CNN and demonstrated competitive results. GMAF has also been used for EEG classification and performed well [150].

The contribution of this chapter is to propose a way to adapt it to multivariate TS by imaging each axis of the sensors separately and fusing them together to create multi-channel images. The resulting image size equals the sequence length of the original TS, which also represents a limitation, this was tackled by employing a linear interpolation on the TS sequence to increase or decrease it. The datasets used in Chapter 2 were employed to compare the accuracies of the two approaches, as well as the smartphone-based recognition of human activities and postural transitions dataset.

4.2 Dataset and pre-processing

In the first part of this chapter, the smartphone-based recognition of human activities and postural transitions dataset from Reyes-Ortiz et al [151] is used. It contains data from experiments that were carried out with a group of 30 volunteers within an age bracket of 19-48 years who performed a protocol of ADL. Six dynamic activities from the dataset were included: walking, walking up, walking down, sit to stand, stand to sit, lying. The reason for choosing these activities is that post-stroke patients are required to perform them in their daily lives. In addition, the quantity of data for the different activities are very close allowing us to build a more accurate model less prone to bias. Besides some of these activities are very similar and hard to differentiate which will be a good challenge for our algorithms. The data is comprised of tri-axial linear acceleration and 3-axial angular velocity at a constant frequency of 50 Hz using the embedded accelerometer and gyroscope in a smartphone. The dataset is organised in two folders the first contains unprocessed raw data and the second contains preprocessed data (denoised and decomposed in different time windows and features). In this work, only the raw data were considered.

4.2.1 Data segmentation

After the data of the different activities were loaded into different frames of data, each element at a particular time was labelled depending on which activity was performed. After that, a sliding window method has been employed in order to prepare the data for further processing. A sliding window converts sequential data into different chunks of data with a fixed window size in order to be used in algorithms that require the data to be of a specific structure. In this work, a sliding window of 4 sec (4 sec × 50 Hz = 200 data elements) was chosen to decompose the dataset into different windows of the same size. The label for each data chunk was chosen to be the label that is most recurrent within the segment. Since the activities were performed sequentially, an activity might be cut when composing the different windows. To remedy to this issue, an overlap of 2 sec was introduced, which means that adjacent windows share 50% of the data. The resulting windows are matrices with fixed sizes 200 × 6 with the six columns corresponding to the triaxial accelerometer and gyroscope. Fig 4.1 shows how a sliding window operates. Therefore a static segmentation as the one presented in section 3.2.2 was employed.

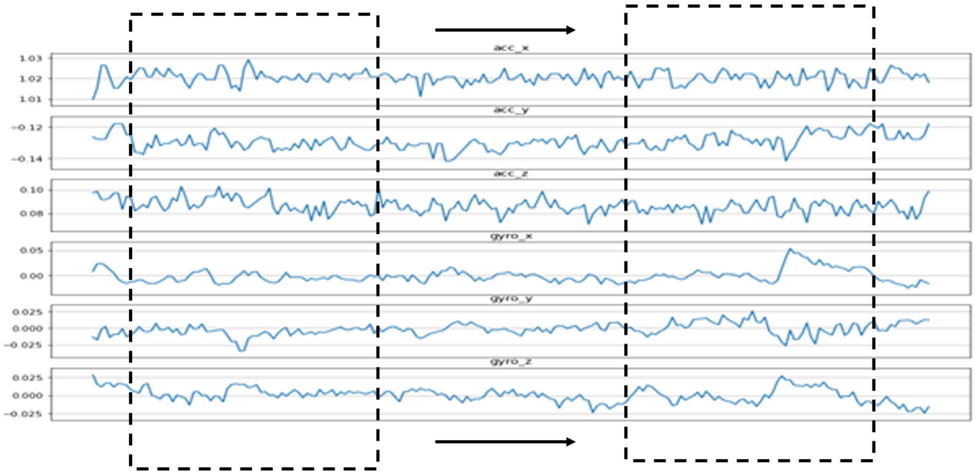


Figure 4.1: Sliding window to decompose the dataset.

4.2.2 Imaging ts data

Three encoding techniques proposed by Wang et.al in [152] are utilised in this chapter namely the GASF, GADF and MKV.

GMAF

GMAF is an encoding technique that transforms TS data into images, it uses the polar coordinates representation of the data presented in a matrix form called the Gramian

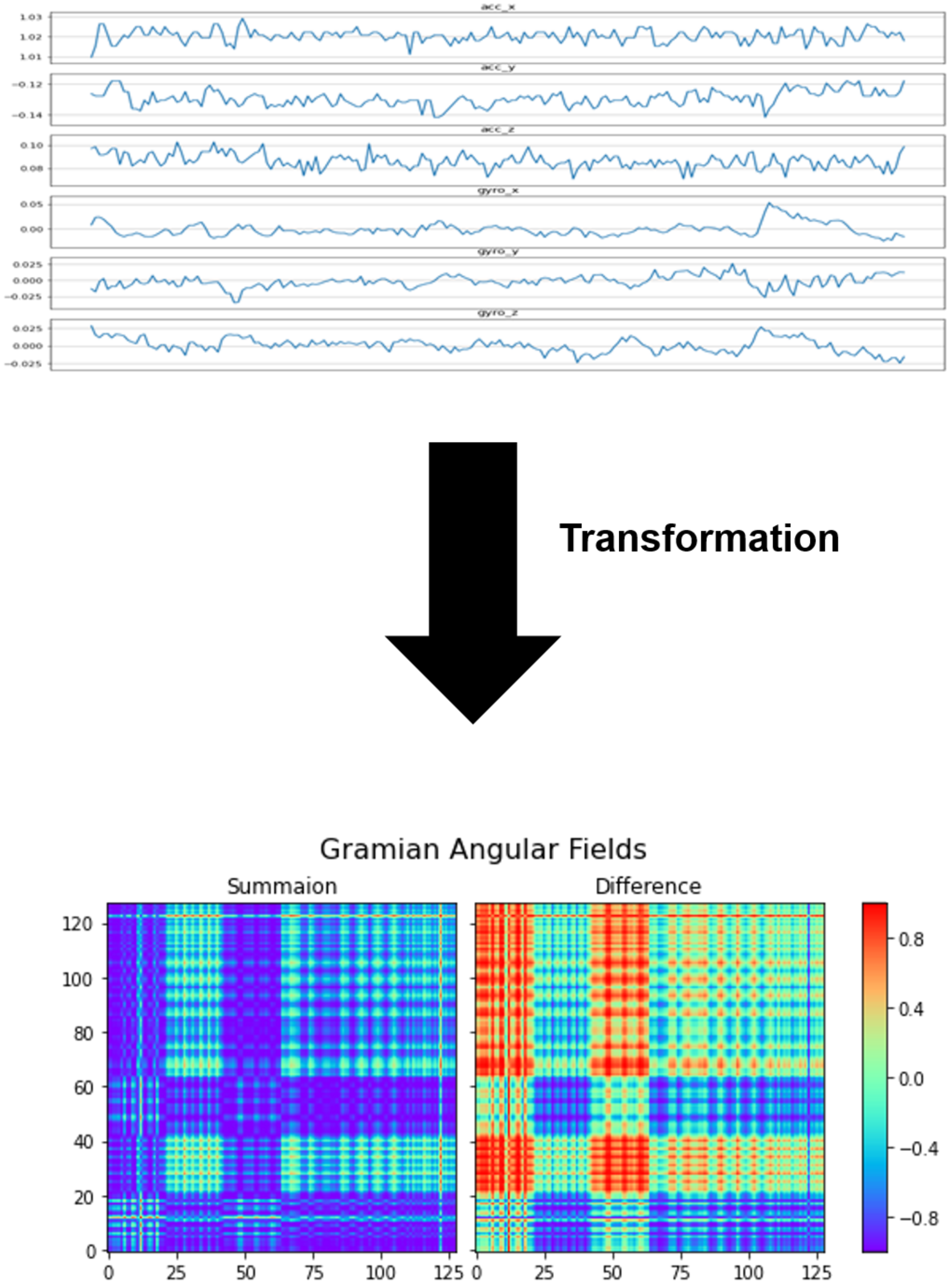


Figure 4.2: Encoding a window of the IMU data into gramian images.

matrix. Each element of this matrix is either the addition (GASF) of the cosines of the polar angles or the difference (GADF) of their sines. This mapping maintains the temporal dependency of the TS, the time increases as the position shifts from the top left to the bottom right. Due to this feature, the polar coordinates can be reverted back to the original TS data by its transformation principle.

The steps to encode the TS data into images using GAF are given below:

- First data should be re-scaled to the range [0,1] (or [-1,1]) using the linear normalisation equation 4.1:

$$\hat{x_i} = \frac{x_i - min(X)}{max(X) - min(X)} \tag{4.1}$$

- After that, the data is mapped into its polar coordinates representation using equations 4.2 and4.3

$$\phi = arccos(\hat{x_i}), -1 \leq \hat{x_i} \leq 1, \hat{x_i} \in X \tag{4.2}$$

$$r = \frac{t_i}{N}, t_i \in \mathbb{N} \tag{4.3}$$

- Finally either sum (GASF) or differentiate (GADF) the cosines or the sines of the polar angles to build the Gramian matrix as shown in 4.4 and 4.5 respectively:

$$\begin{aligned} GASF &= cos(\phi_i + \phi_j) \\ &= \hat{X}^T.\hat{X} - \sqrt{I - \hat{X}^2}^T.\sqrt{I - \hat{X}^2} \end{aligned} \tag{4.4}$$

$$\begin{aligned} GADF &= sin(\phi_i - \phi_j) \\ &= \sqrt{I - \hat{X}^2}^T.\hat{X} - \hat{X}^T.\sqrt{I - \hat{X}^2} \end{aligned} \tag{4.5}$$

where I is the unit vector after the transformation to polar coordinates, X the elements of the TS X, and t the time subscript.

Markov Transition Fields

The MKV is given as follows:

$$M = \begin{bmatrix} w_{ij|x_1 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_1 \in q_i, x_n \in q_j} \\ w_{ij|x_2 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_2 \in q_i, x_n \in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij|x_n \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_n \in q_i, x_n \in q_j} \end{bmatrix} \tag{4.6}$$

A $Q \times Q$ Markov transition matrix (W) is built by dividing the data (magnitude) into Q quantile bins. The quantile bins that contain the data at time stamp i and j (temporal axis) are q_i and q_j ($q \in [1, Q]$). M_{ij} denotes the transition probability of $q_i \rightarrow q_j$. That is, the matrix W which contains the transition probability on the magnitude axis is pread out into the MKV matrix by considering the corresponding temporal positions.

By assigning the probability from the quantile at time step i to the quantile at time step j at each pixel M_{ij}, the MKV M encodes the multi-span transition probabilities of the TS. $M_{i,j||i-j|=k}$ denotes the transition probability between the points with time interval k. For example, $M_{ij|j-i=1}$ illustrates the transition process along the time axis with a skip step. The main diagonal M_{ii}, which is a special case when $k = 0$ captures the probability from each quantile to itself (the self-transition probability) at time step i. To make the image size manageable and computation more efficient, The MKV size is reduced by averaging the pixels in each non-overlapping $m \times m$ patch with the blurring kernel $\{\frac{1}{m^2}\}_{m \times m}$. That is, the transition probabilities are aggregated in each sub-sequence of length m together. To the contrary to the GMAF, This mapping does not maintain the temporal dependency of the TS.

4.3 Classification and experimental results

In subsection 4.3.1 the 2D size images from the GMAF transformations (subsection 4.2.2) were and the windows from the segmentation (subsection 4.2.1 to feed a 1D based CNN classifier, while in subsection 4.3.2 the previous images were converted to RGB format and use a 2D CNN based classifier as well as the VGG_16 pre-trained model employing transfer learning technique and compare the overall results.

4.3.1 2D models

The model used for the classification comprises two 1D CNN layers, supported by a dropout layer for the regularisation of the data, and then a pooling layer. The reason for defining two CNN layers is to give the model a good chance of learning the features from the input. In order to avoid over-fitting of the data resulting from the fast learning of the CNNs a dropout layer is utilised. After the CNN, the features are flattened to a 64-node vector and go through a fully connected layer that provides a buffer between the learnt characteristics and the classification. This model uses a standard tuning of 64 parallel feature maps and a kernel size of 2. The three discussed methods namely: the windowing method, the GASF and the GADF as shown in Fig 4.3 are used as inputs to this classifier.

The results of decomposing the dataset are 7474 different windows of data of 200 samples for the six sensors-axes (7474 × 200 × 6). The encoded images resulting from the Gramian transformation are 7474 of 256 × 256 different images. 80% of the data (5980) were used for training the model while 1494 were used for testing. To evaluate the techniques, the model was used in three separate parts, one for each input technique.

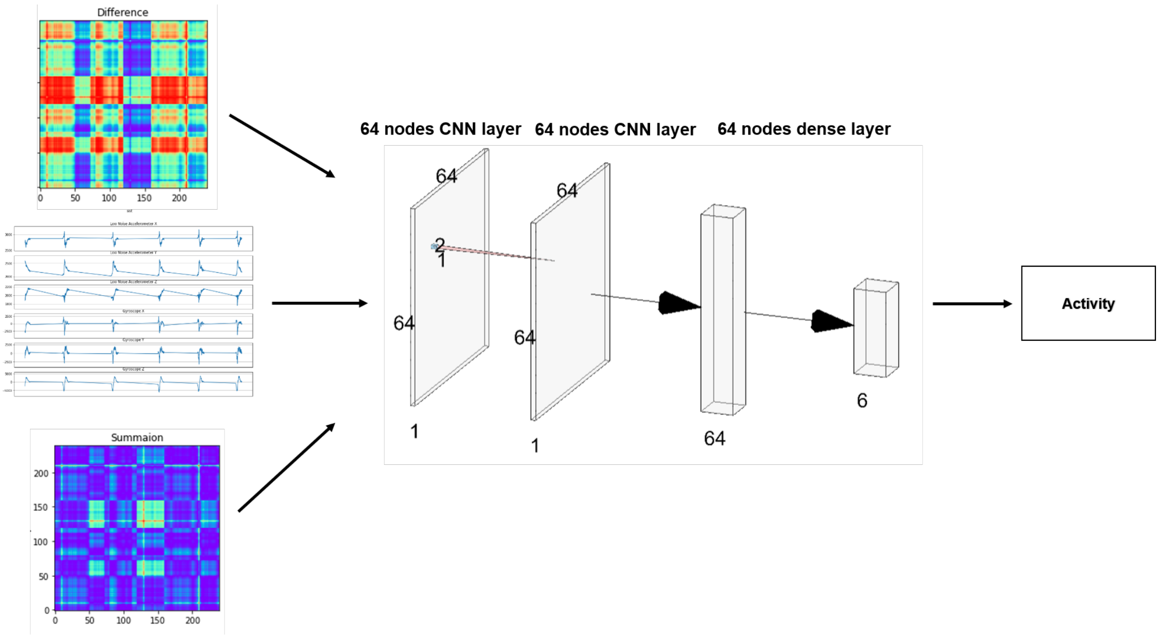
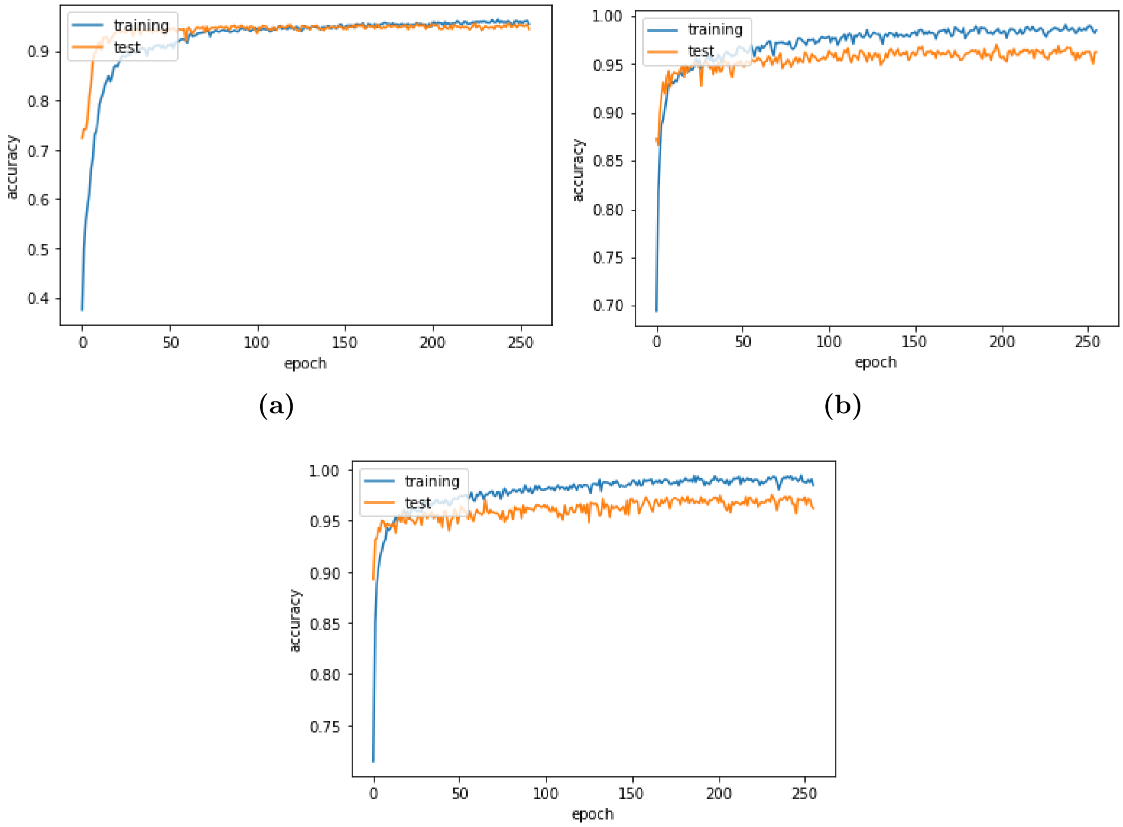


Figure 4.3: Classification process for the 2D methods.

The models were trained for 250 epochs on an I7 CPU 6700T 16GB RAM and the results are shown in Fig 4.4 and 4.5.

- The window-CNN model (Fig 4.4-a) reaches a maximum accuracy of 95.42% for training and 94% for the testing, this model seems less prone to over-fitting as the accuracies seem to stabilise at the same time after 130 epochs at around 94%. This model though starts learning slowly with a training precision of 37.5% and a testing precision of 72.31% at the origin. The average learning time was 740 μs per sample.

- The GASF-CNN model (Fig 4.4-b) reaches a maximum training accuracy of 98.81% and testing accuracy of 97.06% but it seems to start overfitting after 30 epochs. The accuracies seem to stabilise at an accuracy of 98.54% for training and 96.25% for validation. The model also starts learning quickly with a training accuracy of 69.46% at the origin and 87.29% for the testing. The average training time was 770 μs by sample.

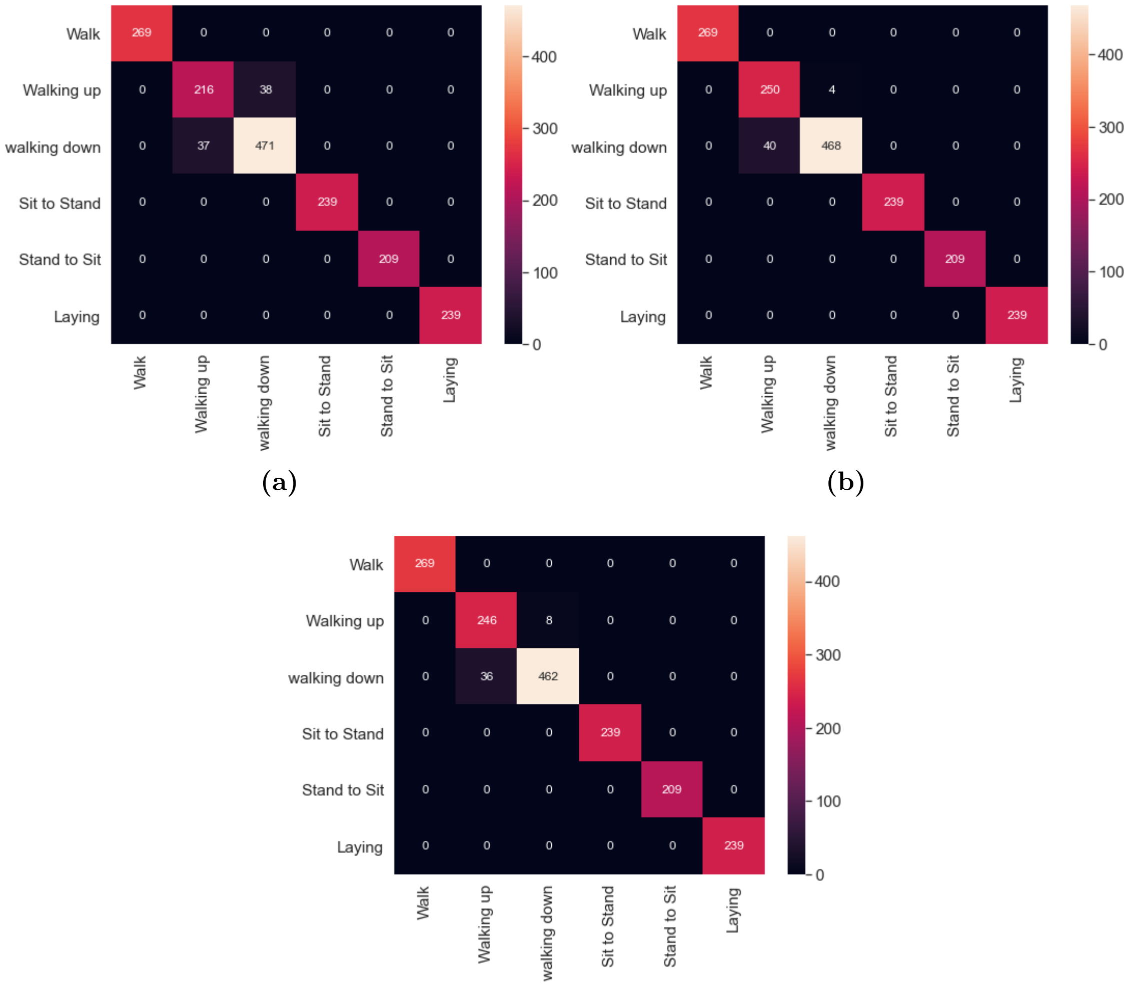


(a)

(b)

(c)

Figure 4.4: Accuracies of the different 2D method



(a)

(b)

(c)

Figure 4.5: Confusion matrices of the different 2D method

- Finally, the GADF-CNN model Fig 4.4-c) reaches a maximum training accuracy of 99.38% and testing accuracy of 97.06%, this model seems to start overfitting after 35 epochs. The accuracies seem to stabilise at 98.43% for training and 96.19% for validation were obtained. This model though starts learning very quickly with a training precision of 71.47% and a testing precision of 89.23% at the origin. The average learning time was 820 μs per sample.
- 75 time chunks from the overa1l 1494 were miss-classifed in the window-CNN (Fig Fig 4.5-a) model. It confuses 38 walking up activities for walking down and 37 walking down for walking up.
- For the GMAF models, 44 miss-classifications for both models were recorded. The GSAF_CNN (fig Fig 4.5-b) confused 40 walk-ups for walk-downs while the GDAF_CNN (Fig Fig 4.5-c) miss-classified 36 walking for walking-down.

4.3.2 RGB models

The 2D: 128 × 128 GMAF images were converted to the 128 × 128 × 3 RGB format in order to investigate their performances using:

The first model comprises 2 layers of 2D CNN 64-nodes supported by dropouts to reduce over-fitting. the learned features are flattened and then filtered out through a 64-node vector to finally go through the Softmax classification layer. This model uses a standard tuning of 64 parallel feature maps and a kernel size of 2 × 2.

In the second model, transfer learning is used by employing the popular VGG16, which is a 16-layer network built by Oxfords VGG [153]. It was pre-trained on a 1,000,000 images dataset from ImageNet and achieved state-of-the-art results. It contains 16 hidden layers composed of convolutional layers and max pooling. One extra Softmax 6-layer classification layer was added at the top for our classification.

As for the 2d models, 80 per cent of the data (5980) were used for training the model while the 1494 were used for testing. To evaluate the techniques, the model was used in four separate structures (depending on the two inputs and the two classifiers) as shown in Fig 4.6. Fig 4.7 shows the models' accuracies when trained for 250 epochs on Google Colab GPU 16GB Ram.

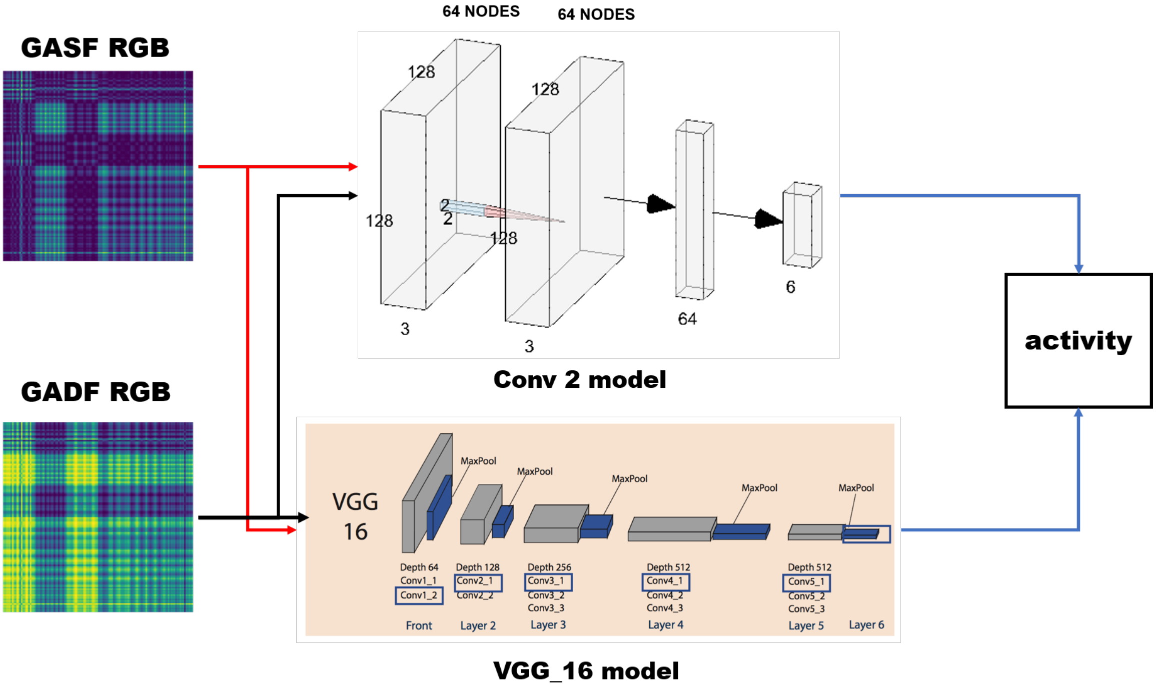
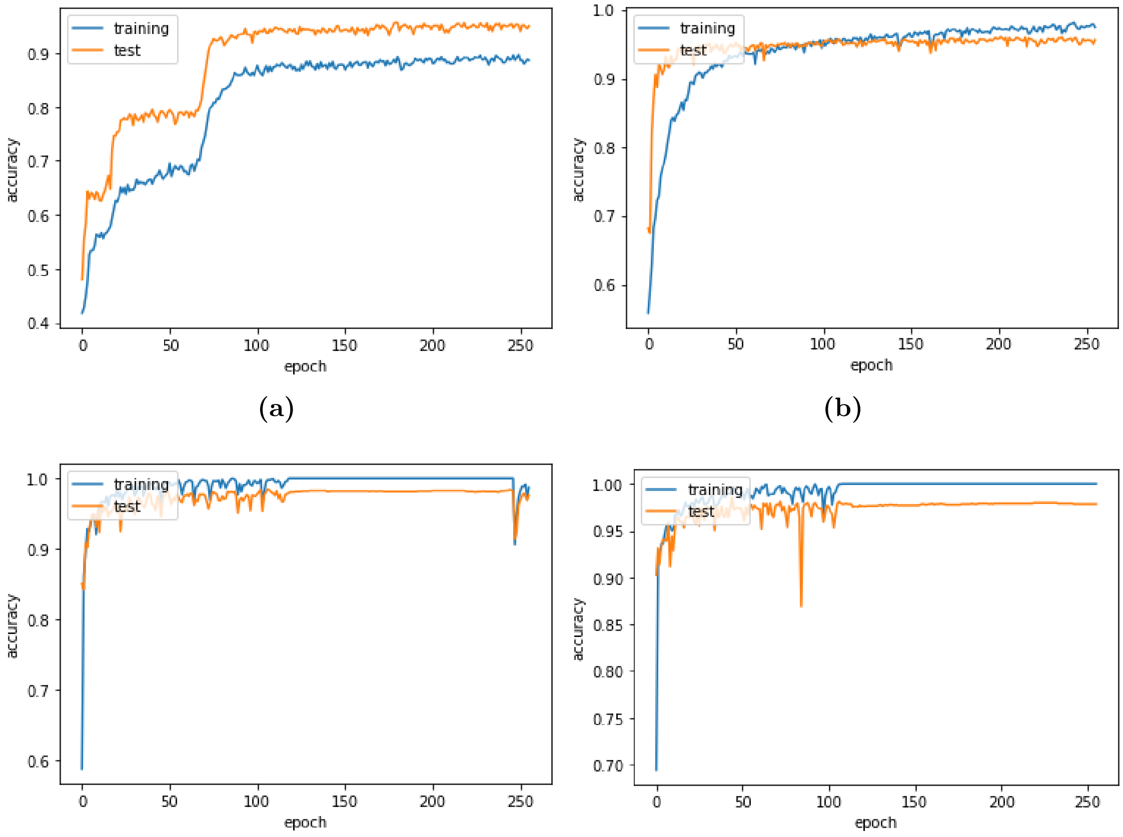


Figure 4.6: Classification process for the RGB methods.

- The CONV2D_GSAF model performs relatively badly (Fig 4.7-a), it reaches a maximum accuracy of 89.53% for training and 95.45% for the testing, this model seems less prone to over-fitting as the accuracies seem to stabilise at the same time after 120 epochs around the accuracies given before. This model though starts learning slowly with a training precision of 41.80% and a testing precision of 47.96% at the origin. The average learning time was 22.33 ms per sample.

- The CONV2D_GDAF model (Fig 4.7-b) reaches a maximum training accuracy of 97.98% and testing accuracy of 97.52% but it seems to start overfitting after 120 epochs. The validation accuracy seems to stabilise at an accuracy of 95.65% while the training keeps increasing above 97.98%. The model also starts learning



(a)

(b)

(c)

(d)

Figure 4.7: Accuracies of the different RGB methods

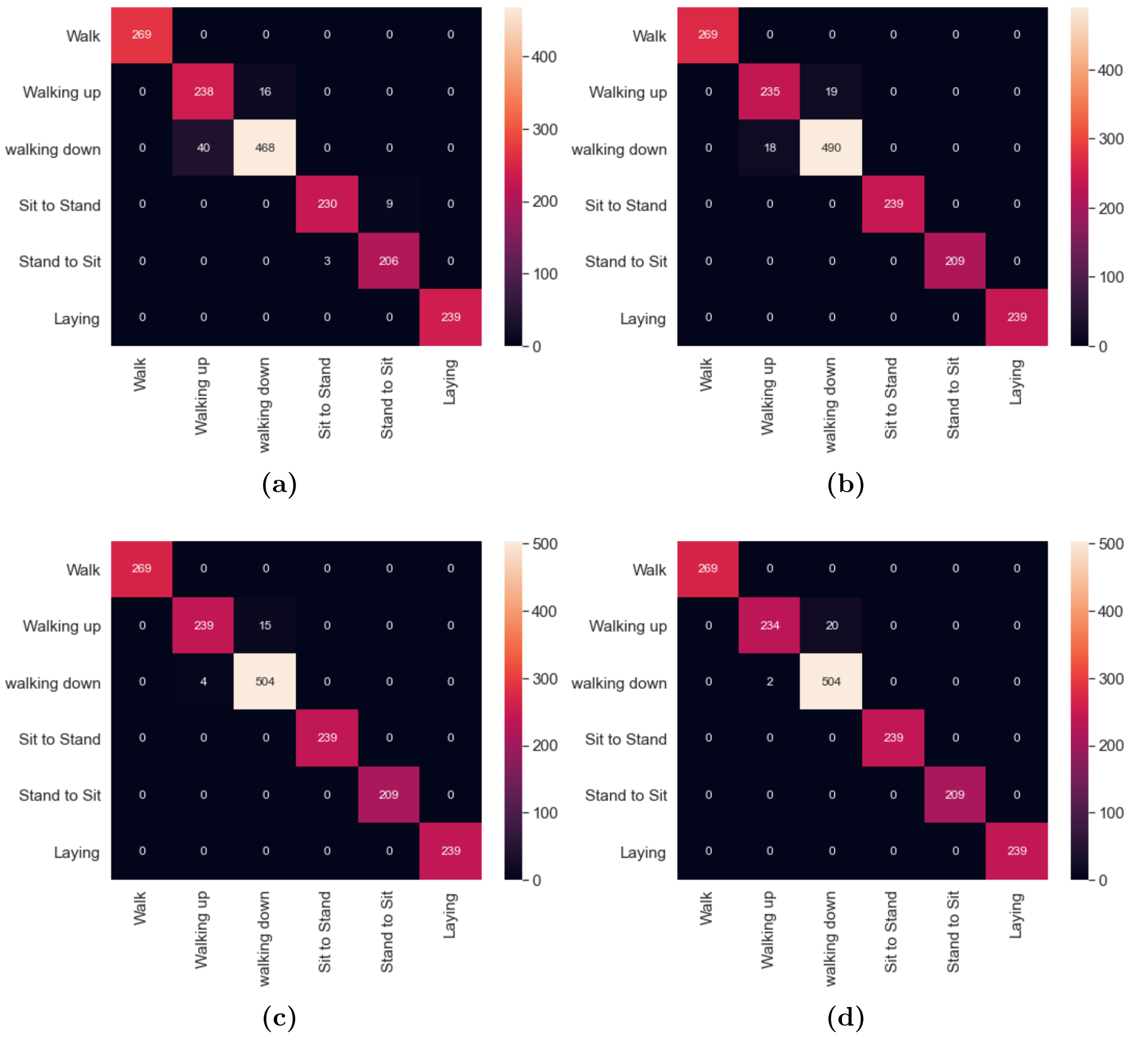


Figure 4.8: Confusion matrices of the different RGB methods

quickly with a training accuracy of 55.88% at the origin and 68.16% for the testing. The average training time was 42 *ms* by sample.

- The VGG_GSAF model (Fig 4.7-c) reaches a maximum training accuracy of 100% and testing accuracy of 98.46%, this model seems to start overfitting after 115 epochs. The accuracies then decrease to accuracies of 98.73% for training and 97.59% for validation. This model though starts learning very quickly with a training precision of 58.86% and a testing precision of 85.08% at the origin. The average learning time was 73 *ms* per sample.

- Finally, the VGG_GDAF model (Fig 4.7-d) reaches a maximum training accuracy of 100% and testing accuracy of 98.53%, this model seems to stabilise after 120 epochs at 100% training and 97.86% for validation. This model though starts learning very quickly with a training precision of 69.41% and a testing precision of 90.23% at the origin. The average learning time was 79 *ms* per sample.

- For the 2D CNN models, 68 and 37 miss-classifications were recorded for the 2D_GSAF (Fig 4.8-a) and 2D_GDAF (Fig 4.8-b) models respectively. The first one mostly confuses walking up and down but also some sit-to-stand and stand-to-sit activities. the second one is more accurate only miss-classifying some walking up and down activities.

- For the VGG models, 19 and 22 miss-classifications for the VGG_GSAF (4.8-c) and VGG_GDAF (4.7-d) models were recorded respectively. The VGG_GSAF) confused 15 walk-ups for walk-downs while the VGG_GDAF miss-classified 20 walking for walking-down.

To summarise, the four RGB-based models give even better accuracies than the 2D models. Using GSAF and the CNN_2D improved the accuracy of the windowing method by approximately 1.45% for the validation data, and decreased the training data by 5.89% for training data but took much longer for training. The reason for that is that the windows of data were encoded to 128 × 128 images and then to RGB 128 × 128 × 3 images.

Using GADF and the CNN_2D improved the windowing accuracy 3.56% for the validation data, and 2.52% for training data, nevertheless the required time for training was slower than the GSAF_CNN2D (almost double). On the other hand the VGG models gave the best results overall, it improved the windowing accuracy by 4.58% for the training data for both GSAF and GDAF, and 4.46%, 4.53% for the test data accuracy for GSAF and GDAF respectively. The time used for the training was the slowest among all models.

following these results, the RGB images are used in the upcoming parts.

4.4 Encoding pipeline

In this section the results obtained in Chapter 3 are compared with the proposed approach which consists of images resulting from encoding the TS data, the images of the different sensors' axes are fused together and image sizes are increased using a linear interpolation technique. The resulting images are then fed to a 2D CNN-based model. The datasets presented in sections: (3.2 and 3.3) are used.

4.4.1 Encoding techniques

4.4.2 Image fusion and interpolation

The TS windows are encoded into images using the previously described encoding techniques in section 4.2.2, images resulting from different axes are fused together to create multiple-channel images. A linear interpolation is used before that in order to either up-sample or down-sample the images' sizes.

Image fusion

These encoding techniques described in 4.4.1 transform univariate TS windows into single-channel images whose size equals that of the window length. in this work, the window size is 200, subsequently, the resulting images are 200 × 200. In the case of multivariate TS, as it is in this work, each sensor's axis is encoded into images. For acceleration or gyroscope-only data, the three axes $(x - y - z)$ are transformed into three-channel images, meaning, the resulting images for the different axis corresponding to the same window are fused together to create three-channel images (A 3×200 window generates $3 \times 200 \times 200$ images). In the case of the combination of the two sensors, six-channel images are generated by fusing the images for each axis (a 6×200 window generates $6 \times 200 \times 200$ images).

Figure 4.9 illustrates two windows of data corresponding to three-axis acceleration data for jogging activity and standing activity and the corresponding GASF, GADF, MKV.

A limitation when using these three encoding techniques is that the resulting image size is set by the window length of the TS data. Sometimes changing the image size would improve classification accuracy, or speed up the training. In this work, an additional layer of pre-processing is done on the TS data to change its size before encoding it into images using linear interpolation.

Linear interpolation

Linear interpolation is a technique to fit a curve using linear polynomials to generate new data points within the interval of a discrete set of already known data points. It has been used in this work to over-sample windows of data to increase the size of the windows by adding new points. Another approach would have been to increase the window size when segmenting the dataset, but that would decrease the number of the resulting windows, hence reducing the dataset size. In addition, as stated earlier, in

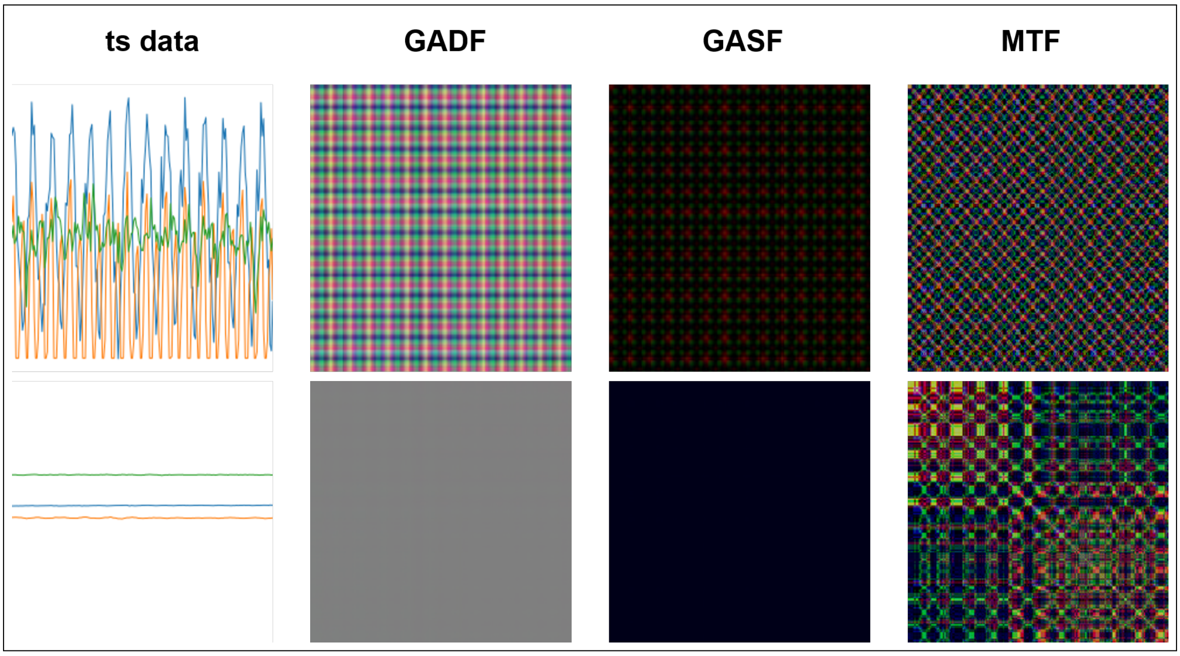


Figure 4.9: The TS data from accelerometer and the different corresponding encoded images for jogging at the top and standing soup at the bottom.

this work the same window used in other works has been chosen in order to have an objective performance comparison.

If the two known points are given by the coordinates (x_0, y_0) and (x_1, y_1), the linear interpolation is the straight line between these points. In this work, an interpolation Coeff is defined, which is a parameter that controls the factor at which the resulting data length (rdl) is increased from the original data length: $Coeff = rdl \div odl$. For example, a coefficient of 2 would yield to doubling the data points of the window. The added data points are homogeneously spread along the window in order not to affect the data distribution. A Coeff in the range [0,1] means reducing the window length, for example, a Coeff of 0.5 would yield to decrease the data points of the window by half. Figure 4.10 describes the coeff-based interpolation used in this work before constructing the images.

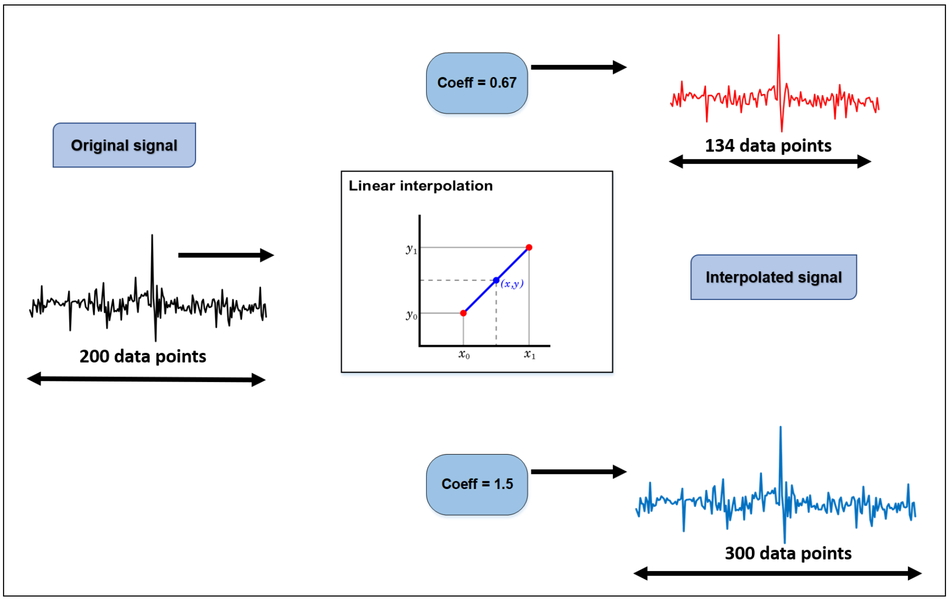


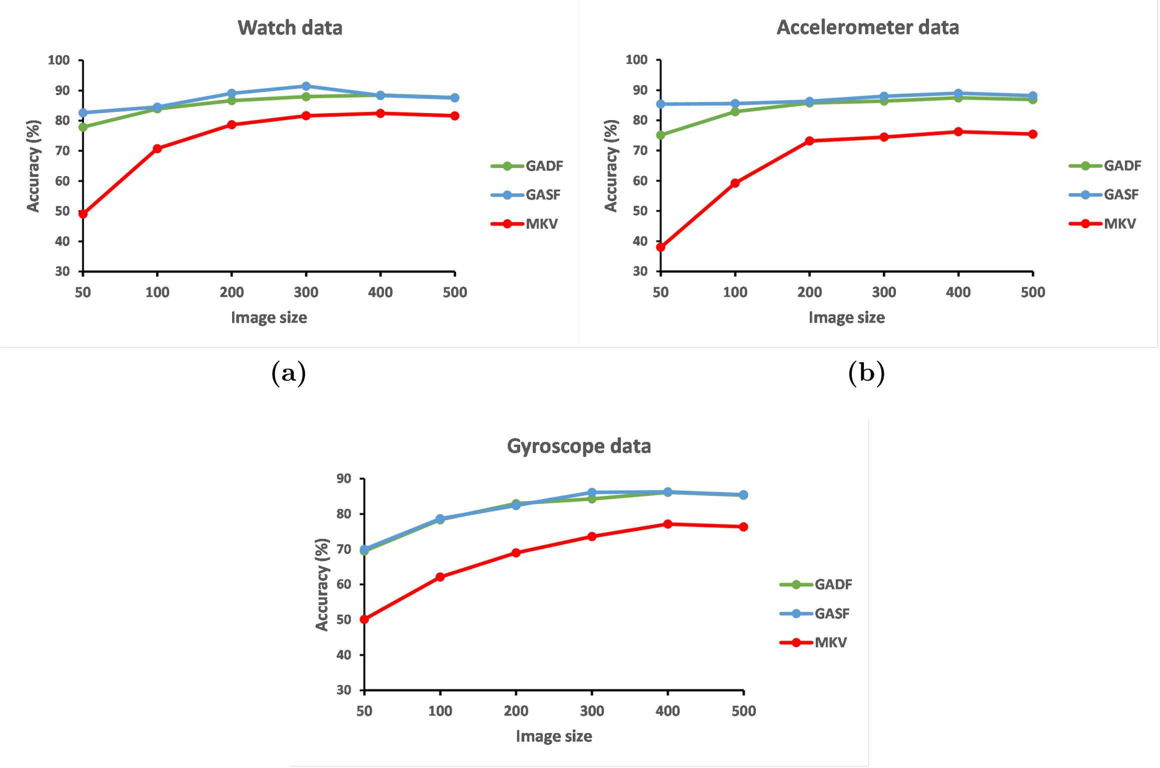
Figure 4.10: Linear interpolation of IMU data.

4.4.3 Data pre-processing and classification model

Only the data from the watch sensors are considered, data from the phone are not included because the sensor placement on the waist did not capture relevant information for hand-oriented activities, thus yielding a worse classification performance.

The resulting 200-long chunks of raw data from section 3.2.2 are normalised and re-scaled to the range $[-1, 1]$. After that, a linear interpolation is done on the windows either to up-sample or down-sample the data windows as described in section 4.4.2 to see the effect of either increasing or decreasing the image size. The sizes chosen are: 50, 100, 200 (no interpolation), 300, 400, 500. after that, the associated windows for each sensor are encoded using three encoding techniques described in section 4.4.1. The resulting images for the different axes are fused together to create 3-channel images when the source is gyroscope only or accelerometer only, or 6-channel images for their combination as explained in section 4.4.2.

The dataset is decomposed into training and validation following the same procedure in



(a)

(b)

(c)

Figure 4.11: Accuracies per image size for the different encoding techniques

section 3.2.2, the fine-tuning parameters are also chosen similarly. Many classification models have been investigated and the Xresnet18 [132] was finally selected because it performed best without transfer learning. Xresnet is the popular Resnet [122] architecture with three different tweaks on the residual blocks.

4.4.4 Experimental results on the WIDSM dataset

The accelerometer, gyroscope data and their combination from the watch are encoded into different-sized images as explained in 4.4.3 and then fed to the XResnet18 model. The resulting validation accuracies are given in Figure 4.11 and Table 4.1, representing the accuracy for each encoding technique for each sensor per image size. From these results, numerous points were noticed:

Table 4.1: Accuracies per image size for the different encoding techniques

Sensor	Encoding	50	100	200	300	400	500
ACC	GADF	75.11327	82.8838	85.775	86.3878	87.4496	86.8796
	GASF	85.3897	85.5596	86.3028	88.0017	88.9573	88.1504
	MKV	38.0123	59.2483	73.2003	74.4653	76.237	75.4301
GYRO	GADF	69.473	78.3815	82.9524	84.2814	86.1418	85.3349
	GASF	70.0093	78.6847	**82.416**	**86.1473**	86.3003	85.4934
	MKV	50.1435	62.1398	69.0065	73.6358	77.1659	76.359
Watch	GADF	77.8624	83.9086	86.646	87.9552	88.4551	87.6482
	GASF	82.6232	84.5275	**89.0502**	**91.4544**	88.3361	87.5292
	MKV	49.0597	70.7451	78.6479	81.6234	82.421	81.6141

- Fusing the images significantly increased accuracy from the TS models in section 3.2.2 from 83% to 91.5% for the combined sensors, GASF encoding 300 image size.

- The GASF is the best-performing encoding technique, it performs slightly better than GADF for all the sensors and sizes.

- MKV is the worst performing encoding, it performs worse than the TS model from section 3.2.2.

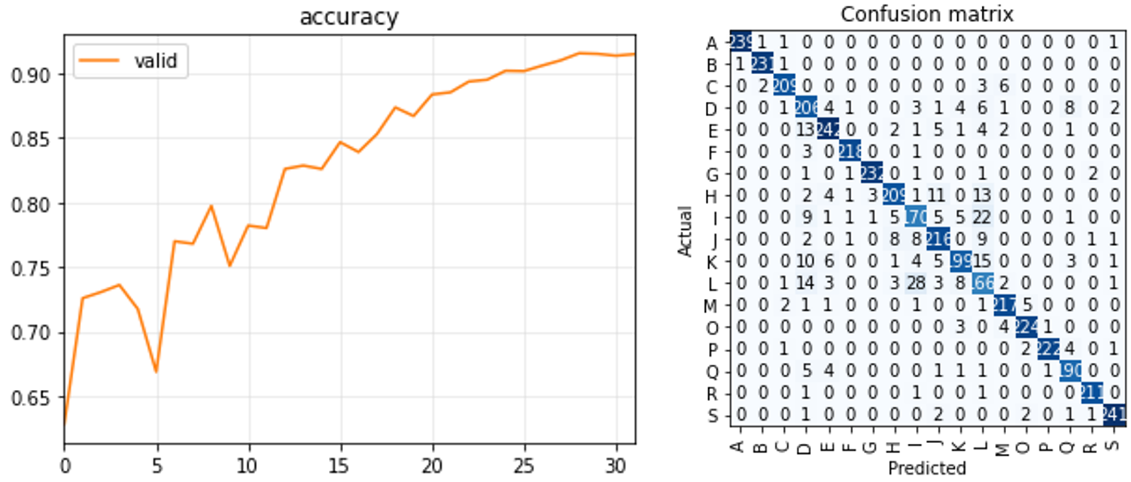
- Up-sample Interpolation of the data increased the classification accuracy for almost all the models until a $Coeff \approx 2$ e.g. It increased the accuracy for the GASF-200-watch from 89.1% to 91.5% for GASF-300-watch, and 82.9% for the GADF-200-gyro to 86.1% for GADF-400-gyro.

- Increasing Coeff further than 2, led the performance accuracy to decrease in all models.

- Down-sample interpolation improves the accuracy from the TS models from section 3.2.2 for some models, a Coeff of 0.5 performed better for example in the GASF-100-ACC, GADF-100-watch, and GASF-100-watch. Only a single model with a Coeff of 0.25, i.e. the GASF-50-ACC performed better than the TS model. Down-sample interpolation can be useful when a performance accuracy improvement is sought, and keeping also a balance with the computation complexity.

- 6-channel images resulting from fusing all the sensors performed better than the 3-channel images from the individual sensors.

Figure 4.12 shows the validation accuracy curve and the confusion matrix for the best-performing model which is the GASF-300-watch.



42

Figure 4.12: Accuracy per epoch and confusion matrix for the GASF 300 size data from the watch.

Comparing the confusion matrices in Figure 4.12 and the TS model in Figure 3.3, demonstrates a significant improvement in distinguishing between the hand-oriented activities, this shows that by interpolating the data and fusing the images the model learnt to discriminate between these very close complex activities.

The model achieves near-perfect classification for most of the activities, the most confusion happens only between eating a sandwich and eating chips accounts for 25 false negatives each among the 238 (24%). Besides, the general hand activities and non-hand oriented activities miss-classifications have been reduced.

4.4.5 Comparaison with other works

‘

Table 4.2: Comparison with other works

Paper	[154]	[155]	[156]	ExceptionTime	[157]	[158]	Our approach
Accuracy (%)	79	80	72	83	84	87.5	91.5

All the work done on the WIDSM dataset employed the traditional segmentation technique. Benavidez et.al in [154] employed both LSTM and CNN architectures, reaching a maximum accuracy of 79% for the watch. Saleh et.al in [155] opted for using conventional ML algorithms consisting of Random Forest, KNN and SVM. They selected a set of features and achieved a maximum of 80% for RF. Online federated learning techniques were used in [156] achieving an accuracy of 87%. Researchers in [157] used four different DL models namely CNN, BiLSTM, LSTM, and ConvLSTM on the different sensors of the dataset, the CNN model outperformed the other models achieving 84.9%. Finally, Bhuiyan et.al in [158] applied several ML algorithms along with some preprocessing techniques to identify which combination performs best, they found out that the highest accuracy 87.5% is achieved in phone accelerometer data when coupling Principal Component Analysis with Random Forest. Table 4.2 summarises the accuracy result comparison. An inference speed comparison was not possible as these data were not available in the other works' papers.

The proposed model performed better than all other existing works, achieving 91.5%. The best-performing model was increasing the data from the combination of accelerometer and gyroscope from the watch using linear interpolation by a factor of 1.5, then encoding it into GASF, to finally feeding the resulting images into the Xresnet18 model.

4.5 Conclusion

The segmented TS data were converted into images using three distinct encoding techniques, namely GASF, GADF, and MKV. The resultant images from the various axes of the sensors were merged to produce multi-channel images. To increase the dimensions of the images, linear interpolation was applied to the data windows to generate additional data points. The optimal configuration was determined by comparing the performance of the various encoding techniques and image dimensions, and evaluating how these parameters impacted the classification accuracy. The classification accuracy was improved by up to a factor of 2 by increasing the image dimensions using linear interpolation; however, the accuracy declined beyond this point. Additionally, reducing the image dimensions from the original window length enhanced the accuracy compared to TS models, striking a balance between performance and computational complexity. Merging the images boosted the accuracy from 83% for 1D models to 91.5% in the case of GASF encoding and 300-size images. MKV encoding, on the other hand, performed poorly when compared to the other encoding techniques.

The proposed methodology in this chapter significantly improved accuracy compared to Chapter 3 and could, therefore, be a suitable approach for post-stroke patient HAR and addressing the performance limitations identified in Section 2 of Chapter 2.

In the upcoming chapter, we will delve into the limitation highlighted in section 4 of Chapter 2, specifically focusing on the issue of data quantity. The chapter will explore the use of GANs to create varied and high-quality time series data within the realm of post-stroke rehabilitation. This exploration aims to enhance the assessment process in post-stroke rehabilitation further. This will also contribute to the algorithms proposed in this chapter by providing more data.

Chapter 5

Data augmentation

5.1 Background

In sensor-based telerehabilitation, data plays a crucial role as it serves as the medium used to learn features. The data generated by these sensors are known as TS [159], which are organized sequentially in a time-dependent manner. They are classified as univariate TS if they vary on a single axis and multivariate TS if they vary on multiple axes [111]. However, in healthcare applications, including stroke rehabilitation, it is not always feasible to obtain sufficient data due to patients' conditions that may prohibit their attendance in testing sessions [160]. To address this issue, the creation of synthetic data, with enough information to simulate real-world data, is required. This process, known as data augmentation, is a well-established processing step in CV [161]. Additionally, data augmentation assists with model generalisation capability, reducing over-fitting and increasing the trained models' characterisation boundary [162]. Although many data augmentation techniques exist for TS data, such as random transformations like scaling and slicing, they are not always effective as they cannot account for the specific characteristics of each dataset [163].

To overcome this challenge, new ML models have been introduced that allow personalised spawning of data by taking into account the input dataset's characteristics, such as GANs. GANs are DL models that capture the inner probabilistic distribution of actual data and generate new comparable data. However, GANs suffer from the issue of mode collapse, where the generated data do not account for all the elements of the real dataset, resulting in synthetic data that fails to learn all the information from the real one.

The contributions provided in this chapter are summarised as follows:

- A new GAN model was proposed by coupling it with a Siamese network (SN), to add another layer that allows to generate more heterogeneous data.
- The resulting model generates more diverse TS than the original GAN, as proved using the longest common sub-sequence (LCSS). Classifying the original data using the generated TS increased from 63% in the original GAN to 98.2% in the proposed model, for the first dataset and from 48% to 90.8% in the second.
- Encoding TS into images permitted to increase the classification performance, thus improving the post-stroke tele-rehabilitation assessment.

5.2 Datasets

This study utilises two datasets that belong to the post-stroke rehabilitation categories. Section 5.2.1 examines an ARAT-based dataset introduced by Lee et al. in [151]. Section 5.2.2 explores the WISDM Smartphone and Smart-watch Activity and Biometrics Dataset proposed by Weiss in [126]. These datasets were chosen because they belong to the categories of assessment systems identified in section 3 of Chapter 2 namely: Activity recognition and movement classification for the WISDM dataset and Clinical assessment emulation for the ARAT dataset.

5.2.1 ARAT dataset

The dataset includes ARAT motions [164], which are rated on a four-point scale. A score of 3 indicates satisfactory completion within 5 seconds, while a score of 0 denotes non-completion due to factors such as inability to grasp the cube or use fingers to manipulate it. The score also considers the time taken to complete the task, where a score of 2 indicates completion with difficulty or taking an abnormally long time, and a score of 1 represents partial completion. The trial involved 34 stroke patients undergoing rehabilitation over a 60-day period in a hospital setting. Each patient performed a set of ARAT motions up to three times in a single session, with data continuously recorded and manually segmented into individual trials. Notably, the scores were awarded on a session basis, suggesting averaging over trials, and multiple therapists scored the sessions, introducing score variability despite briefings. Each class used in this study comprises a distinct number of segments, and the length of each segment varies, as presented in Table 5.1. The data acquisition is sampled at 30 Hz.

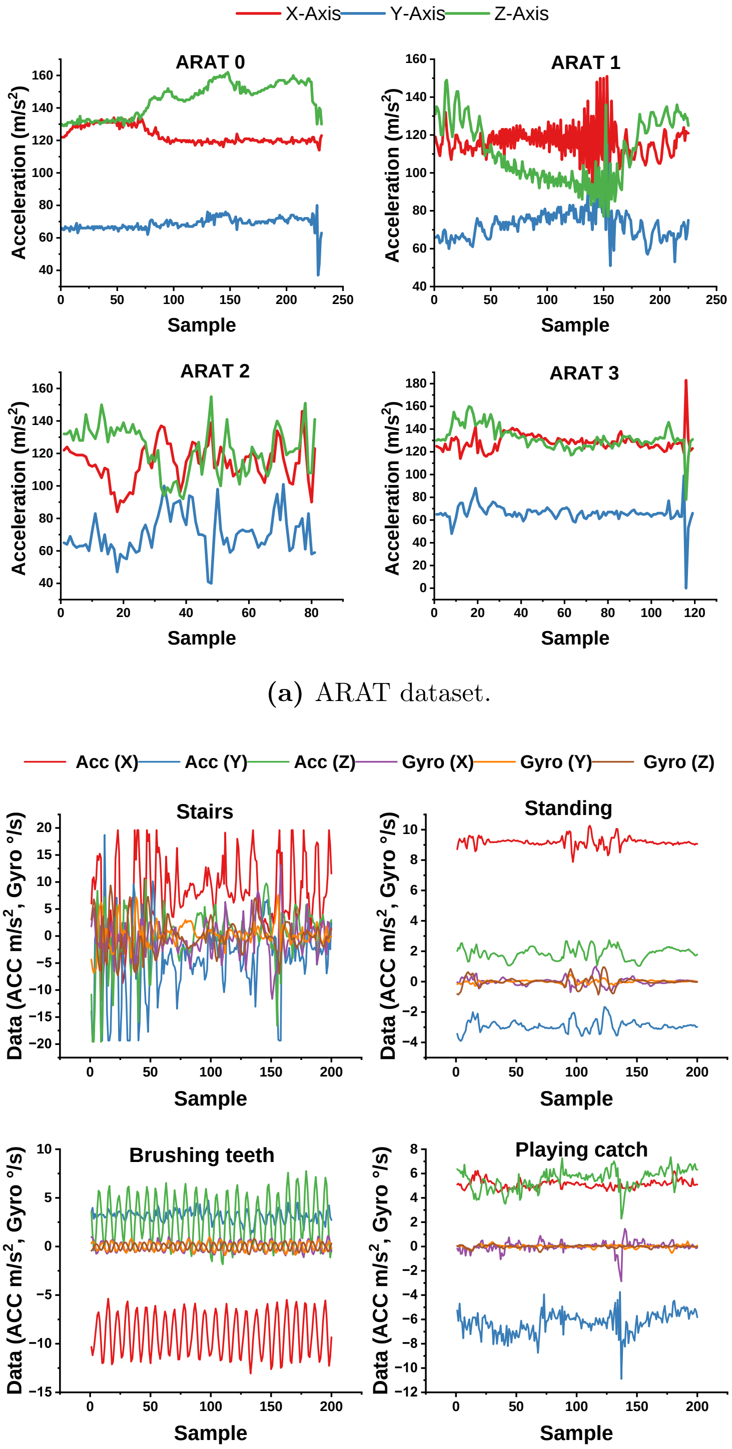
Table 5.1: Number of TS segments per class in the ARAT dataset.

ARAT score	0	1	2	3
Number of segments	3	6	38	31

As depicted in Table 5.1, the dataset is unbalanced and contains an insignificant number of segments for contemporary algorithm analysis such as deep learning. This feature renders the dataset an excellent candidate for the exploration of generating synthetic data. Furthermore, the dataset is comprised of naturalistic data recorded in a real-life scenario, thereby offering added value to its application in real-world scenarios. The segments are multivariate TS chunks of varying lengths, derived from triaxial acceleration as previously mentioned. Each TS in a particular class has dimensions of $n \times 3 \times t$, where n represents the number of segments in that class, 3 indicates the number of axes (X, Y, Z), and t denotes the sequence length.

5.2.2 Activity recognition dataset

The WISDM dataset introduced in 3.2.1 is used.



(a) ARAT dataset.

(b) WISDM dataset.

Figure 5.1: *Some of the original TS segments for the datasets.*

Figure 5.1 displays samples from each dataset, Figure 5.1a displays some segments from each ARAT class and Figure 5.1b shows segments from four of the activity recognition datasets. Time axis “sample" refers to the acquisition index knowing that the first dataset has 30 acquisitions per second while the 2^{nd} has 20.

5.3 2D systems

In this section, the process of generating synthetic data from the original dataset using a GAN is presented. The aim is to produce data that satisfies two key conditions: first, the generated data must be realistic and accurately represent the statistical distribution of the original dataset. Second, it should not suffer from mode collapse. The proposed GAN structure is described in detail and demonstrates the generated synthetic data in the following sections.

5.3.1 GAN

The proposed GAN is composed of two parts: the first part encompasses a generator that takes as its input random noise vectors z and generates dummy data while the second part “the discriminator" takes the real TS data and the dummy TS data generated by the generator as input, and outputs a number that is corresponding to the probability of the input being real. The GAN employs the Nash equilibrium game principle [165], which assumes two players. The generator tries to learn the real TS data distribution whereas the discriminator tries to accurately guess whether the fed data is from the original dataset or from the generator. To be victorious in the game, the two players shall compete repeatedly to improve both the generation (maximise resemblance) and the discrimination (minimise the difference).

Mathematically, let $\mathbf{x}$ be a TS window from the dataset distribution p_X, and $\mathbf{z}$ be a random vector. only $\mathbf{z}$ from a uniform distribution with a support of $[-1, 1]$ is considered, Let G and D be the generative and discriminative models, the generative model takes $\mathbf{z}$ as input and outputs the TS data, $G(\mathbf{z})$, that has the same support as $\mathbf{x}$. Denote the distribution of $G(\mathbf{z})$ as p_G. The discriminative model approximates the probability that the input TS data is drawn from p_X. Ideally, $d(\mathbf{x}) = 1$ if $\mathbf{x} \sim p_X$ and $D(\mathbf{x}) = 0$ if $\mathbf{x} \sim p_G$. The generative and discriminative models can be trained together by solving the equation (5.1) below:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{real}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}))] \quad (5.1)$$

In our work, the architecture for the two parts is only comprised of fully connected layers. Depending on the dataset, two architectures were proposed:
The first dataset utilises a generator with five layers, beginning with an input layer that corresponds to the latent vector z of 32 elements. This is followed by a fully-connected layer of 256 nodes, and three additional fully-connected layers of 512, 1024, and 699 nodes, respectively. Each layer is equipped with a Leaky ReLU activation function and batch normalisation is applied. Finally, the last layer is reshaped to a 3×233 node configuration that matches the architecture of the input data. The discriminator, on the other hand, takes as input either real or artificial data that is first reshaped to an 899-node fully connected layer. This layer then passes through two additional fully connected layers of 512 and 256 nodes, respectively, both of which are equipped with leaky ReLU activation functions and batch normalisation. The output node is a single node with a Sigmoid activation function, responsible for indicating whether the data is real or artificial.

For the second dataset, the same architectures are used, with only differences being a latent vector size of 128 nodes and the inclusion of an additional fully connected layer of 2056 nodes after the 1024 layer. This is followed by a 1200-node layer that is reshaped to 6 × 200 instead of the 899-node layer in the previous dataset. The discriminator for this dataset begins with a 6 × 200 input that is flattened to 1200 nodes, followed by three additional fully connected layers of 512, and 256, and 128 nodes, respectively.

The architectural details of these setups are summarised in Table 5.2.

Table 5.2: Architecture of the proposed GANs for the two datasets.

Dataset	Network architecture	Layer type	Nodes, activation, normalisation
ARAT	Generator	Input vector	32
		Fully Connected Layer	256, Leaky ReLU, Batch normalization
		Fully Connected Layer	512, Leaky ReLU, Batch normalization
		Fully Connected Layer	1024, Leaky ReLU, Batch normalization
		Fully Connected Layer	699, Leaky ReLU, Batch normalization
		Reshape Layer	3 x 233
	Discriminator	Input layer	3 x 233
		Fully Connected Layer	512, Leaky ReLU, Batch normalization
		Fully Connected Layer	256, Leaky ReLU, Batch normalization
		Output layer	1, Sigmoid
Activity recognition	Generator	Input vector	128
		Fully Connected Layer	256, Leaky ReLU, Batch normalization
		Fully Connected Layer	512, Leaky ReLU, Batch normalization
		Fully Connected Layer	1024, Leaky ReLU, Batch normalization
		Fully Connected Layer	2056, Leaky ReLU, Batch normalization
		Fully Connected Layer	1200, Leaky ReLU, Batch normalization
		Reshape Layer	6 x 200
	Discriminator	Input layer	6 x 200
		Fully Connected Layer	512, Leaky ReLU, Batch normalization
		Fully Connected Layer	256, Leaky ReLU, Batch normalization
		Fully Connected Layer	128, Leaky ReLU, Batch normalization
		Output layer	1, Sigmoid

5.3.2 Generated data from the GAN model

ARAT dataset

It has been observed in Section 5.2.1 that the dataset is imbalanced and has a limited number of TS segments. This makes training the deep neural networks challenging. To address this issue without the need for additional data collection, a GAN-based data augmentation technique has been proposed to generate synthetic data. This approach can help overcome the problem of imbalanced data, and its applicability may extend to other TS-related research studies.

As TS segments have varying lengths, direct application of deep learning algorithms is not feasible. These algorithms require input streams of equal length, which can be achieved by padding the segments with zeros to match the length of the longest segment which is 233 samples (time acquisition). This approach is a well-established technique in TS analysis and has been used in several studies as a simple and effective way to achieve equal-length input data [166]. Moreover, in [167], zero-padding was used in their GAN-based approach for generating synthetic TS data to handle variable-length TS data.

After padding, the segments were normalised to the range of [0, 1]. Unlike image generation, where the label of the generated image can be visually recognised, in the TS domain, it is difficult to associate each generated window of data with its corresponding real-domain counterpart. Therefore, a separate GAN has been trained on each class of the data, with each class trained separately, and the data for each class generated independently.

Generating synthetic data separately for each class also avoids potential biases that could arise from the padding process. By training the GAN separately for each class, the generated data will have the same padding and distribution as the original data for that class. This is important because the padding process may introduce a bias in the generated data if performed on the entire dataset together.

After conducting multiple trials with different latent vector sizes, a size of 32 data points was determined to be the most suitable. The criterion for determining the optimal latent vector size was based on the quality of the generated data, as increasing the latent vector size beyond 32 did not significantly improve it.

Adam optimiser [168] with a momentum of 0.5 and learning rate of 2×10^{-4} are empirically selected for both the generator and discriminator red by testing various GAN models and trials, and binary cross-entropy was used for their compilation. The selection of the learning rate was empirical and involved testing various values. The aim was to identify a value that would strike a balance between convergence speed and overfitting avoidance. The choice of using the Adam optimizer with a momentum of 0.5 and binary cross-entropy for compilation is a common practice in many GAN architectures. This decision was also informed by empirical testing.

For each class, 640 epochs were required for training. This number was selected by monitoring the degradation of the discriminator loss. Moreover, Google Colab with 32 GB RAM and a Nvidia T4 GPU was used, moreover, the code was developed in Keras and TensorFlow.

Figure 5.2a showcases several synthetic TS segments generated from different ARAT categories using the proposed GAN model. The generated data share the same $n \times 3 \times t$ dimensional structure as the input data, with X, Y, and Z axes representing triaxial data. The generated TS segments exhibit curvature patterns similar to those observed in the real data segments, as can be seen in the plot. The GAN model can generate multivariate TS data from any input segment, as evidenced by the triaxial data generated by the model. Furthermore, the model has learned to generate the zeros that were padded to the segments to achieve equi-length time windows, as demonstrated in the rightmost plot of Figure 5.2a. A thresholding method was employed to determine the end of the signal, where the amplitude decreases below a specific level. This method involved identifying three consecutive data points with amplitudes equal to or less than a threshold value of 0.05 on all axes.

However, during experimentation, the GAN model did seem from visual inspection to exhibit mode collapse, which caused it to generate synthetic data for some portions of the input segments while neglecting others.

Activity recognition dataset

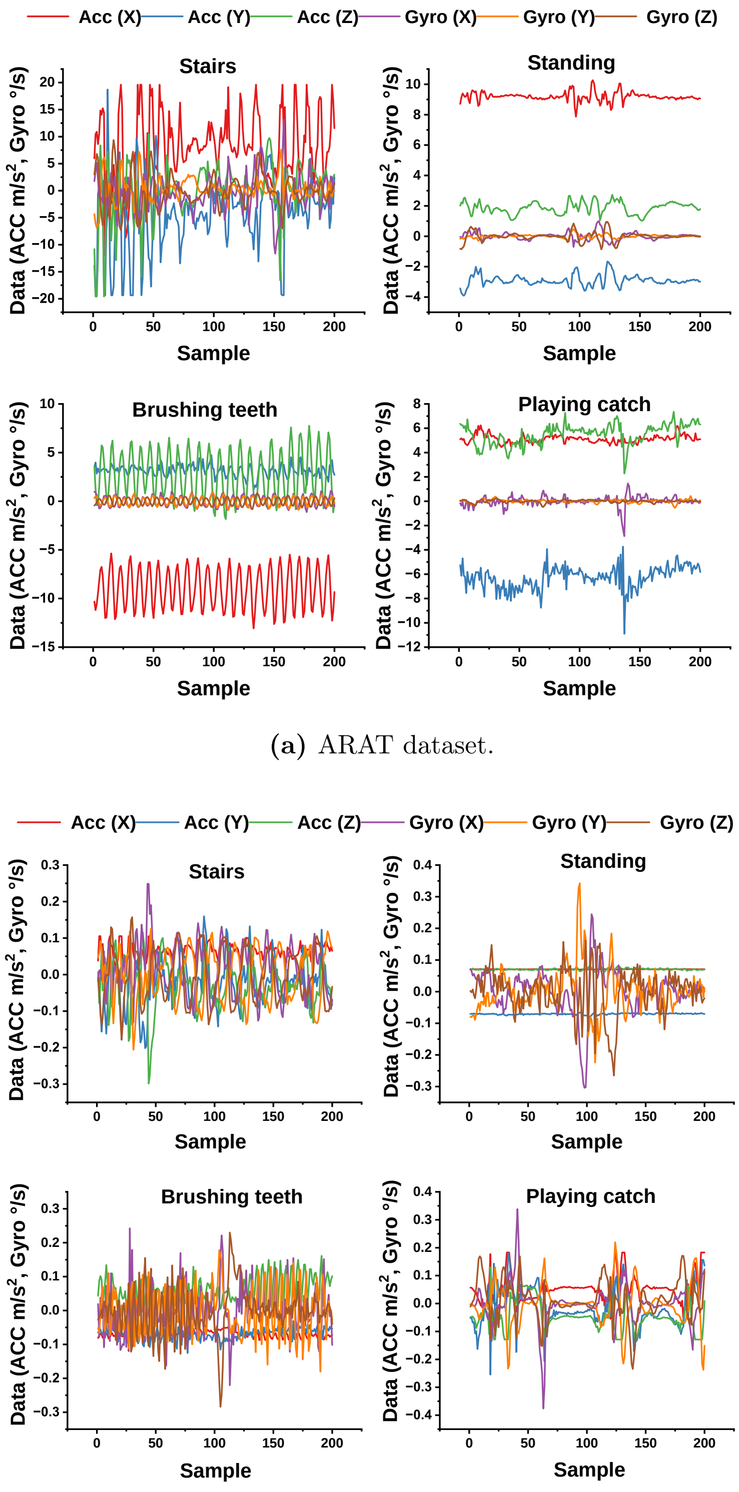
In accordance with the methodology outlined in Section 5.2.2, the dataset was partitioned into 10-second segments consisting of 200 readings each using non-overlapping sliding windows. Subsequently, each data segment was labelled with the activity label that appeared most frequently, resulting in a structured dataset of dimensions $16854 \times 6 \times 200$. In order to generate synthetic data from this dataset, a GAN model was employed as detailed in Section 5.3.1. Prior to applying the GAN, the TS segments were normalised. The hyperparameters used in the GAN training process were similar to those employed previously, with a latent vector size of 128, the Adam optimizer with a momentum of 0.5 and learning rate of 5×10^{-4}, and binary cross-entropy. The GAN training process was repeated for a total of 20k epochs. Figure 5.2b displays the TS segments generated by the GAN model, with each class visually identified through comparison with the real dataset. It is worth noting that the generated data appear to exhibit curvature patterns that resemble those observed in the real data segments. However, it is also observed that some classes of the generated segments appear to have been omitted, indicating a potential occurrence of mode collapse.

While visual inspection can be informative, it is not sufficient to draw accurate conclusions. Further required analysis is presented below.

5.3.3 Analysis of GAN Generated data

The effect of mode collapse is verified on both datasets using two techniques:

1. A similarity study is conducted between the generated segments and the original ones. This is achieved by correlating the generated signal with every original signal and calculating the similarity between them using an objective method. The original signal with the highest similarity is considered the parent signal that spawned the segment. To ensure the robustness of the objective method against



(a) ARAT dataset.

(b) WISDM dataset.

Figure 5.2: *Some of the generated TS segments for the datasets.*

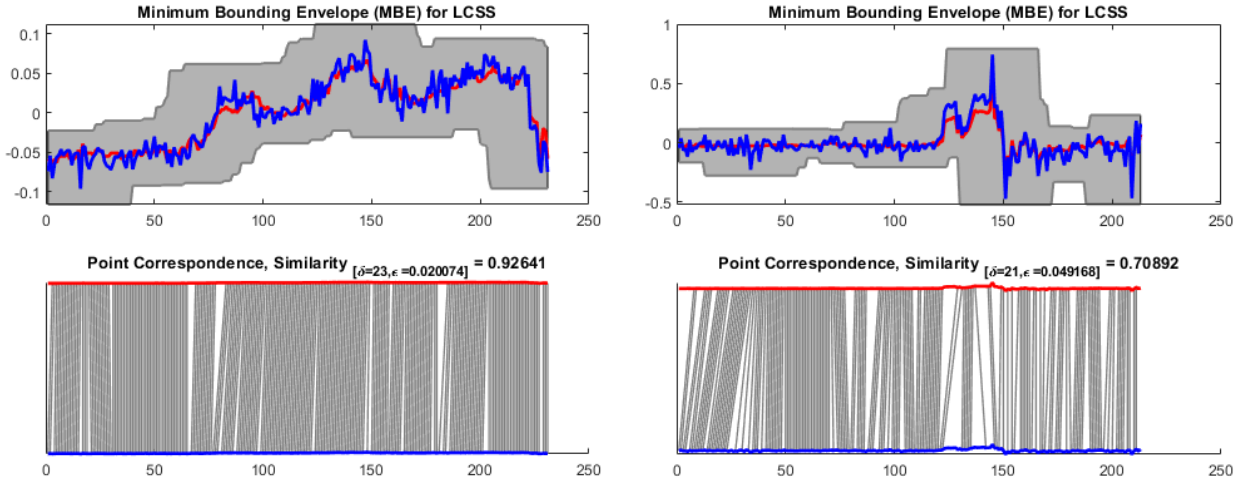


Figure 5.3: LCSS applied to two spawned TS data.

noise or small vibrations, either Dynamic Time Warping or the LCSS [169] are considered. LCSS has been proven to be more resilient under noisy conditions and can work with data of different lengths, hence, it has been chosen for this study. Then, the number of generated data for each class is computed to show the distribution of the generated dataset over the parent classes.

2. The generated data is used to train a classifier, and its performance is evaluated on the real dataset used as a validation set. Classification and other metrics are computed to show whether the generated data contains sufficient information to differentiate between the classes of the original dataset. Before classification, the TS data is encoded into images using Gramian Angular Field (GMAF) and fed into a ResNet-18 classifier. This pipeline has been shown to yield promising results for TS classification of the same dataset in previous works. [170–172].

LCSS algorithm

One of the very first applications of LCSS algorithm has been for string matching [173]. Later contributions worked on the extension of LCSS and it has been widely used for measuring the similarity of two TS with different lengths focusing on similar parts between two TS [173–175].

The basic core method of LCSS is dynamic programming that applies similarity-based searching from machine regions both in time and space to keep away from distant or degenerating regions.

For LCSS operation, let's define **a** and **b** as finite discrete TS. a_1^p is associated with the first TS as **a** with a discrete time index varying between 1 and p. In a similar manner, b_1^q is associated with the second TS **b** with a discrete-time index varying between 1 and q. Additionally, a_i , b_i represent the i^{th} sample of TS a and b, respectively. A recursive algorithm has been formulated to provide a solution to the LCSS [169] as given in Equation (5.2):

$$\mathrm{LCSS}_{\delta,\epsilon}(\mathbf{a}_1^p, \mathbf{b}_1^q) = \begin{cases} 0 & \text{if} \quad p < 1 \quad or \quad q < 1, \\ 1 + \mathrm{LCSS}_{\delta,\epsilon}(\mathbf{a}_1^{p-1}, \mathbf{b}_1^{q-1}) & \text{if} \begin{cases} d_{LP}(a_p, b_q) < \epsilon \quad and \\ \mid p - q \mid < \delta, \end{cases} \\ Max \begin{cases} \mathrm{LCSS}_{\delta,\epsilon}(\mathbf{a}_1^{p-1}, \mathbf{b}_1^{q}) \\ \mathrm{LCSS}_{\delta,\epsilon}(\mathbf{a}_1^{p}, \mathbf{b}_1^{q-1}) \end{cases} & otherwise \end{cases} \tag{5.2}$$

where p and q represent the lengths of TS a and b, respectively, meanwhile, d_{LP} $(a_p - b_q)$ take any L_P -norm of the $(a_p - b_q)$.

Two parameters are used in LCSS to introduce flexibility in controlling the matching regions in time (δ) or space (ϵ). In the end, the similarity of the two times-series is measured using the output of the LCSS including a normalising factor associated with the lengths of input times-series as shown in:

$$S_{\delta,\epsilon}(\mathbf{a},\mathbf{b}) = \frac{\text{LCSS}_{\delta,\epsilon}(\mathbf{a}_1^p,\mathbf{b}_1^q)}{min(p,q)} \tag{5.3}$$

Based on the above definition, the returned values by the LCSS vary from 0 to 1, the highest value is related to a situation when the two TS fully match, and vice-versa. The values of δ and ϵ are taken from the work in [169], which concluded that their best values are:

$$\epsilon = 0.5 \times (min(std(a), std(b)) \tag{5.4}$$

where std is the standard deviation of a and b,

$$\delta = round(0.1 \times n) \tag{5.5}$$

where n is $min(length(a,b))$

Figure 5.3 shows two different generated signals (blue) and their associated parent signal (red) and the corresponding similarity of 0.93 and 0.71 using the LCSS algorithm.

GMAF

The Gramian matrix is a matrix-based encoding method that converts TS data into images by using polar coordinates as a representation of the data. Each component of this matrix is either the addition of the sines of the polar angles (GASF) or the difference of their cosines (GADF). The time increases as the location shifts from the top left to the bottom right, thus maintaining the temporal dependence of the TS. This feature allows the polar coordinates to be converted using the transformation principle back to the original TS data.

In this work, the used GASF are summarised as follows:

- First, using the linear standardisation equation, re-scale the data to the range

[0,1] (or [-1,1]). 5.6:

$$\hat{x_i} = \frac{x_i - min(X)}{max(X) - min(X)}, \tag{5.6}$$

- After that, using equations 5.7 and 5.8, the data is translated into its polar coordinates form.

$$\phi = arccos(\hat{x_i}), -1 \leq \hat{x_i} \leq 1, \hat{x_i} \in X, \tag{5.7}$$

$$r = \frac{t_i}{N}, t_i \in \mathbb{N}. \tag{5.8}$$

- Finally, the cosines of the polar angles to get GASF representation are summed as follows:

$$GASF = cos(\phi_i + \phi_j) = \hat{X}^T \cdot \hat{X} - \sqrt{I - \hat{X}^2}^T \cdot \sqrt{I - \hat{X}^2} \tag{5.9}$$

where X represents the components of the TS X, I is the unit vector following the transformation to polar coordinates, and t is the time stamp index. Figure 5.4 shows the GMAF encoded images for the three axes of segments from ARAT 0, ARAT 1, ARAT 2 and ARAT 3 categories.

Results for the ARAT dataset

For the ARAT dataset, 1500 segments are generated for each ARAT category. To determine the parent segment for each generated segment, LCSS, as described in Section 5.2, is used to find the segment with the highest similarity, as explained in subsection 5.3.3. The results for each ARAT class are presented in Tables 5.3, 5.4, 5.5, and 5.6 respectively. These tables highlight the bias in the generation of the data. For instance, Table 5.3 indicates that 97.3% of the generated data corresponds to segment 3, whereas segments 1 and 2 only account for 1.53% and 1.2%, respectively. A similar bias is observed in other classes where certain segments are not generated at all, indicating mode collapse.

Table 5.3: The number of generated segments per class seen by LCSS algorithm for GAN for ARAT 0.

Segment	1	2	3
GAN	23	18	1459

Table 5.4: The number of generated segments per class seen by LCSS algorithm for GAN for ARAT 1.

Segment	1	2	3	4	5	6
GAN	0	1055	445	0	0	0

Table 5.5: The number of generated segments per class seen by LCSS algorithm for GAN or ARAT 2.

Segment	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
GAN	5	0	42	51	12	99	9	2	0	0	0	0	2	26	3	40	40	97	138
Segment(cont)	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
GAN	7	19	17	158	71	44	190	39	3	19	1	75	261	0	1	0	7	0	22

After this, the total generated dataset of 6000 segments (1500 for each class) are used to train the classifier described in Section 5.2. The resulting dataset after the data encoding was 6000 × 3 × 233 × 233 (three channels 233 × 233 images). Figure 5.4 shows GMAF encoded images for ARAT 0 segment.

The images were fed to a pre-trained ResNet-18, and the training process followed a cyclical learning rate as suggested by Smith in [135], which has been proven effective in previous studies [170–172, 176]. The model was trained for 20 epochs using the dataset, and the original data encodings were used as a validation set. Two metrics were employed for evaluating the performance of the model: Accuracy and F1-score weighted by class. The latter metric was chosen as it takes into account the performance of each class.

The F1-score weighted by class is calculated using the following equation:

Table 5.6: The number of generated segments per class seen by LCSS algorithm for GAN for ARAT 3.

Segment	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
GAN	0	28	6	0	0	0	13	3	7	122	10	80	0	51	14	49
TS-SGAN	69	52	26	73	35	65	61	43	12	118	14	65	50	13	44	16
Segment (cont)	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
GAN	24	28	66	55	2	162	191	77	68	29	129	271	0	13	2	

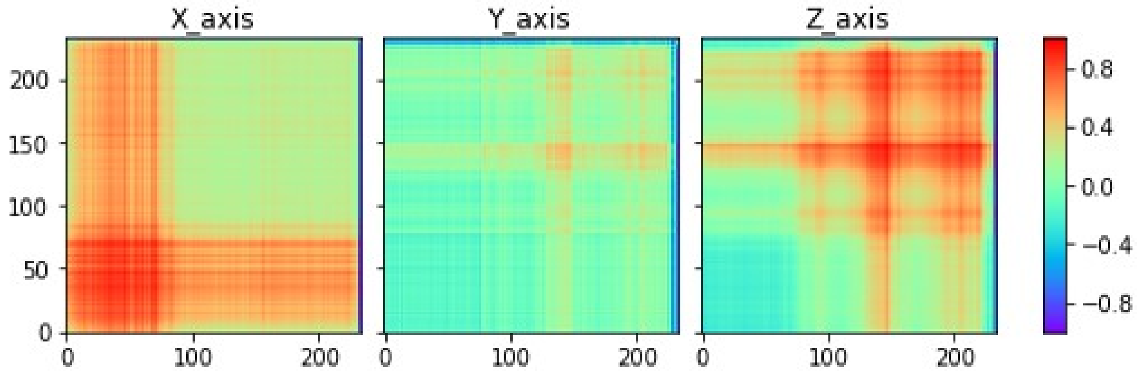


Figure 5.4: An encoding example of an ARAT 0 TS segment into its GASF.

$$F1_{weighted} = \frac{\sum_{i=1}^{n} w_i F1_i}{\sum_{i=1}^{n} w_i}$$

where n represents the total number of classes, w_i is the weight assigned to class i, and $F1_i$ is the F1-score for class i. The weights w_i are defined based on the class distribution in the validation dataset.

Based on the results obtained, it was found that the classification accuracy of the model is 63%, indicating that the model is struggling to accurately classify some of the classes. This is further supported by the F1-score weighted by class of 0.58. These results suggest that the generated dataset used for training the model did not provide sufficient information about the original dataset used for validation and metric computation. The observed mode collapse in the generated dataset may have led to the poor performance of the model on some classes.

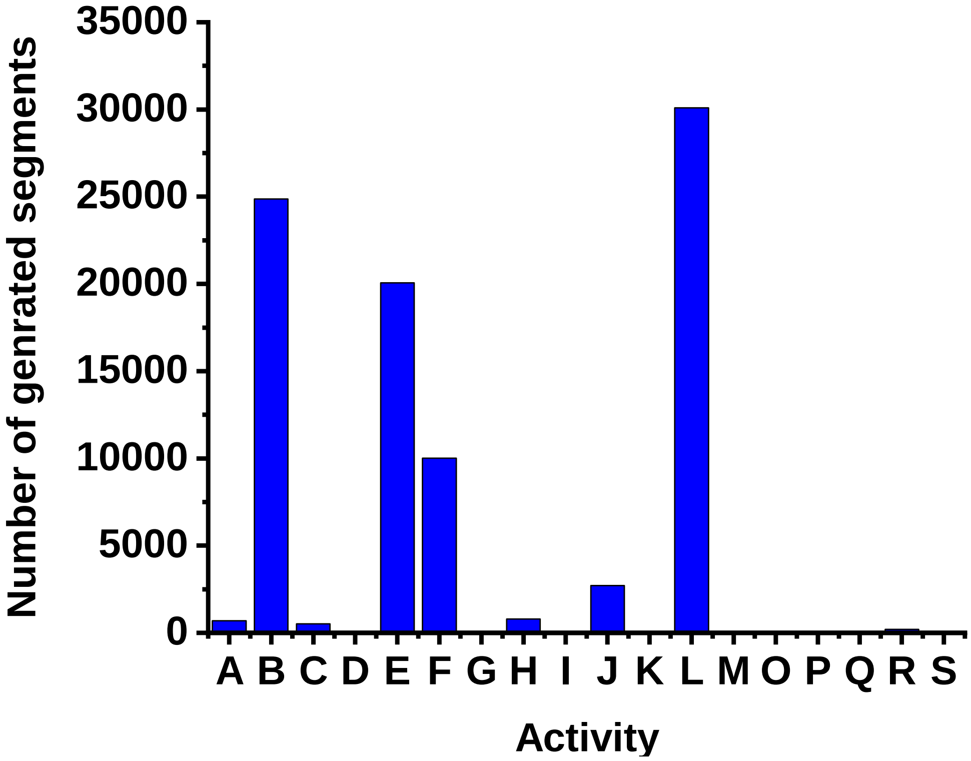


Figure 5.5: Distribution of the GAN generated data per class.

Results for the activity recognition dataset

The second dataset was processed in a similar manner as the first one. A total of 90000 TS (TS) segments were generated using the GAN, and the LCSS method was used to find the similarities with the parent segments. The results are presented in the distribution bar chart shown in Figure 5.5.

From the chart, it is evident that the generated data suffers from a significant bias, with some classes having over 20000 segments while some others have none. This generated dataset was then used to train the same classifier used for the first dataset. A training dataset of 90000 × 6 × 200 × 200 (six channels of 200 × 200 images) was used, with 800 segments from each class taken as a validation set, resulting in a total of 14400 segments. The trained model achieved an accuracy of 48.73% and an F1-score weighted by class of 0.45. These results indicate that similar to the first dataset, the model is experiencing mode collapse and that not all the information has been captured while generating the synthetic data.

To address this issue, incorporating an SN into the GAN architecture is proposed, which is described in Section 5.4.1.

5.4 TS-SGAN to treat mode collapse

The data generated from the GAN in section 5.3 suffers from mode collapse which results in the data not being heterogeneous. To solve this an SN is added, this was inspired by the work in computer vision of Allahyani et al. in [177] and adapting it into the TS domain.

In [178] the SN was initially introduced for use in the tasks involving face and signature verification. Two sub-networks with common weights make up SN [179]. SN compares the characteristics of the pair networks using Euclidean distance while learning the features from each sub-network. As a result, during the training, the network aims to increase the distance between feature pairs (latent data) when they are from separate classes while reducing it when they are from the same class. SNs have been normally employed frequently for the re-identification task because of this attribute as the job's objective is to determine how similar two sequences are to one another [180, 181]. The associated verification loss is given in Equation (5.10).

$$Loss_{ver}(L_i, L_j) = \begin{cases} \frac{1}{2}\|L_i - L_j\|^2 & i = j \\ \frac{1}{2}\max(m - \|L_i - L_j\|, 0)^2 & i \neq j \end{cases}, \tag{5.10}$$

where m is the margin, and L_i and L_j are the latent data for the i^{th} and j^{th} TS data sequence. They correspond the output of the last layers on the i^{th} and j^{th} branch before being fed into the similarity metric function.

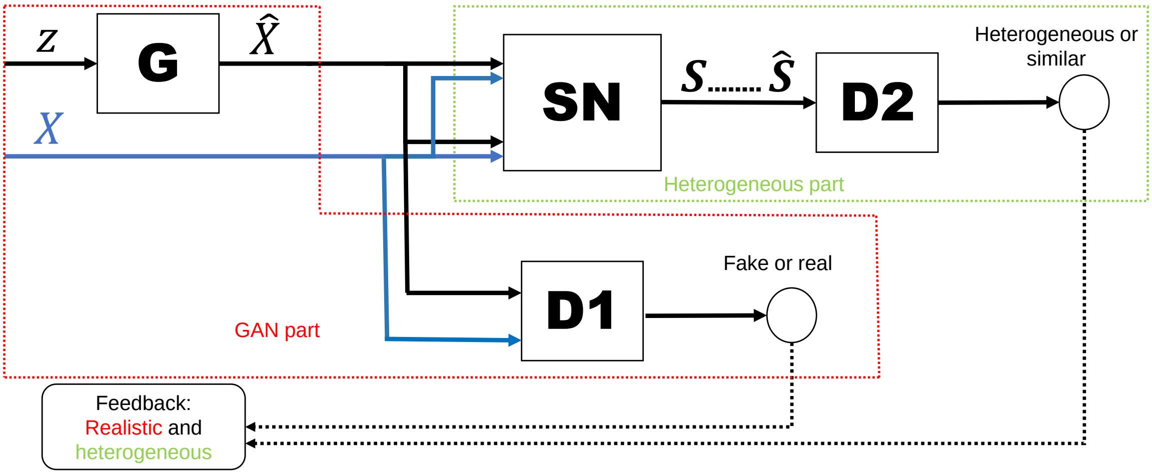


Figure 5.6: Siamese GAN flowchart.

5.4.1 TS-SGAN

Our suggested method for reducing the mode collapse problem is to combine the SN with the modelled GAN architecture to construct a time-series Siamese GAN (TS-SGAN). This will add an additional layer that will learn to differentiate between the different segments of the input layer and try to spawn more heterogeneous data. As shown in Figure 5.6, the TS-SGAN is divided into two components; the first component consists of a generator G and a discriminator $D1$, which are the fundamental configuration in every GAN design. G produces artificial data $\hat{X}$ using random noise vectors z as input. The discriminator $D1$ accepts its inputs from the real data X and the created data $\hat{X}$ so that the networks outputs a probability of the data to be real. The second part of the network is to be responsible for the heterogeneity of the generated data. It is made up of a SN and a $D2$ discriminator. The SN in the TS-SGAN architecture finds similarity in a batch of data, it generates the similarity of the entire batch for real data (S) as well as the created data $\hat{S}$; this serves as an additional layer to spot mode collapse. The inputs to $D2$ are S and $\hat{S}$. If the data in the batch are heterogeneous, the $D2$ identifies the similarity as a true similarity. In any other case, the $D2$ identifies it as a bogus similarity thus indicating mode collapse.

In order to implement the TS-SGAN, the function inherited from the GAN as well as the function responsible for the heterogeneity of the output data are merged. The first part was demystified in section 5.3 and given by Equation (5.1). The latter one, which is responsible for treating mode collapse by varying the GAN output a heterogeneity principle for both G and $D2$, The role of $D2$ is to discriminate between the heterogeneity in the real TS data and the heterogeneity in the created one. Very similar to the Nash equilibrium game principle between G and $D1$ to generate realistic Ts data, the Nash equilibrium between G and $D2$ can be viewed as a means to generate heterogeneous data [182].

The SN in the TS-SGAN takes a pair of real data denoted as p_{real} and outputs their similarity S. Simultaneously, it takes another pair from the generated data denoted as p_{fake} which also generates their similarity $\hat{S}$. The role of $D2$, then, is to discriminate between the heterogeneity of p_{real} and p_{fake}. Finally, G role is to generate p_{fake} TS data that possesses p_{real} heterogeneity. Consequently, G in this part, tries to minimise the cost function while $D2$ tends to maximise it. The process is given in Equation (5.11).

$$\min_{G}\max_{D_2} V(D_2, G) = \mathbb{E}_{\boldsymbol{x1},\boldsymbol{x2}\sim p_{\text{real}}}[\log D_2(SN(x_1, x_2))]+ \\ \mathbb{E}_{\boldsymbol{z1},\boldsymbol{z2}\sim p_z(\boldsymbol{z})}[\log(1 - D_2(SN(G(\boldsymbol{z_1}), G(Z_2))))] \quad (5.11)$$

The architecture of the heterogeneity part of our proposed TS-SGAN is shown in Table 5.7. The SN comprises of two 2D CNN layers with a 3 × 3 kernel including similar padding and *tanh* activation function with successively 4 and 16 channels, followed by a flatten function before outputting the similarity value. $D2$ comprises a simple MLP layer of 128 nodes with the output node responsible for generating either bogus for mode collapse or real for the heterogeneous data.

Coupling this last part with the GAN part from section 5.3 gives us the TS-SGAN

Table 5.7: Siamese Network and Discriminator 2 architectures.

Network	Layer Type	Number of Nodes, Activation Function, Batch Normalization
Siamese Network Architecture	Input Layer	3 x 233 or 6 x 200
	Convolutional Layer	4, Tanh, Same Padding
	Convolutional Layer	16, Tanh, Same Padding
	Flatten Layer	11184
	Output Layer	1, Linear
Discriminator 2 Architecture	Input Layer	3 x 233 or 6 x 200
	Fully Connected Layer	128, Leaky ReLU, Batch Normalization
	Output Layer	1, Sigmoid

architecture. The corresponding overall Loss function of the TS-SGAN is given in equation (5.12).

$$
\begin{aligned}
\min_{G}\max_{D_1,D_2} V(D_1,D_2,G) = {} & \mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}(\boldsymbol{x})}[\log D_1(\boldsymbol{x})] \\
& + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D_1(G(\boldsymbol{z}))] \\
& + \mathbb{E}_{\boldsymbol{x1},\boldsymbol{x2}\sim p_{\text{real}}}[\log D_2(SN(x_1,x_2))] \\
& + \mathbb{E}_{\boldsymbol{z1},\boldsymbol{z2}\sim p_z(\boldsymbol{z})}[\log(1 - D_2(SN(G(\boldsymbol{z_1}),G(Z_2))))]
\end{aligned} \tag{5.12}
$$

5.4.2 Algorithmic process

Figure 5.6 shows the flowchart of the proposed TS-SGAN, as discussed earlier, it comprises G and $D1$ for the GAN part responsible for generating realistic data, and SN and $D2$ responsible for generating heterogeneous data. G takes the latent data vector Z as input and outputs the bogus TS data, while, $D1$ takes instances of real TS data and the bogus TS data, and outputs the probability that the input is real. The SN takes two batches from a dataset DTS that contains both bogus $\hat{X}$ and real data X and outputs the similarity between them ($\hat{S}$ for bogus data and S for real data). Finally, $D2$, takes S and $\hat{S}$ and produces the probability of heterogeneity.

Hence, the steps to produce heterogeneous and realistic data using the TS-SGAN are:

- The SN is trained on the real dataset, to learn to differentiate between the segments.
- A batch of latent data z is fed to G in order to generate bogus data $\hat{X}$.
- Real data is divided into two parts X_1 and X_2 and are fed to SN in order to produce S.
- Bogus data is divided into two parts $\hat{X}_1$ and $\hat{X}_2$ and are fed to SN in order to produce $\hat{S}$.
- $D2$ takes S and $\hat{S}$ to produces heterogeneity probability and $D1$ takes X and $\hat{X}$ to produces how real probability.
- The process is repeated until p_{bogus} converges to p_{real}, and the parameters of SN, G, $D1$ and $D2$ are updated according to the loss function provided in Equation (5.12).

The pseudo-code of the TS-SGAN is given in Algorithm 1.

5.4.3 Experimental results

In this section, the experimental results of our proposed TS-SGAN model are presented, which was trained using the procedure described in Section 5.4.1 and Section 5.4.2. Specifically, the pre-processing and configuration used for training the GAN were also used for training TS-SGAN. Moreover, the same analysis was conducted in Section 5.3.3, which involved finding the parent segment for the generated data using LCSS and training the same classifier on the generated data, followed by using the real data as a validation set.

Algorithm 1 TS-SGAN Algorithm to produce realistic heterogeneous TS data

```
Input      : Dataset of real TS data and the number of iterations as $I$ for SN.
Output     : G that produces realistic and heterogeneous TS data.
Parameters : $\delta$, $\theta$, $\Gamma$ which are parameters of $D1$, $D2$, $G$ , are consecutively initialised.
Sample of noise data $[Z_1.....Z_m]$ .

Sample of bogus TS data $[\hat{X}_1.....\hat{X}_m]$.

DTS sample containing both bogus$\hat{X}$ and real data X.

while $iterations < I$ do
    Train SN on DTS $(\hat{X} \cup X)$
end
while $\hat{X}$ not converge to $X$ do
    Generate Z $[Z_1.....Z_n]$

    $G(Z)$ random batch of latent TS data $[\hat{X}_1.....\hat{X}_n]$

    X random batch of real TS data $[X_1.....X_n]$

    Split latent TS in two: $\hat{X}^1 = [\hat{X}_1.....\hat{X}_{\frac{n}{2}}], \hat{X}^2 = [\hat{X}_{\frac{n}{2}}......\hat{X}_n]$

    Split real TS in two: $X^1 = [X_1.....X_{\frac{n}{2}}], X^2 = [X_{\frac{n}{2}}......X_n]$

    $S = SN(X^1, X^2), \hat{S} = SN(\hat{X}^1, \hat{X}^2)$
    Update $\delta$, $\theta$, $\Gamma$ using Equation (5.12).
end
```

ARAT dataset

Initially, the SN was trained for 32 epochs to acquire the ability to distinguish between the distinct segments contained in the dataset. After that, the hyperparameters employed in Section 5.3.3 for GAN training were utilised, to train the GAN namely, 640 epochs, the Adam optimizer with a momentum of 0.5, binary cross-entropy loss function, a latent vector size of 32, and a learning rate of 2×10^{-4}.

Upon conducting a visual inspection of the generated data, it was observed that the spawned time chunks exhibited a comparable quality to those generated by the model trained in Section 5.3.1. Notably, the model learned to generate the padded zeroes that were added to ensure uniform signal length, which is a consequence of retaining the GAN component of TS-SGAN. However, it is evident that the TS-SGAN model effectively incorporates all dataset segments in generating novel data. Furthermore, a thorough visual examination reveals that the model does not suffer from mode collapse, this is substantiated below.

Following the same procedure used in Section 5.3.1, 1500 segments per class are generated, and results are shown in Tables 5.8, 5.9, 5.10, and 5.11 respectively.

Table 5.8: The number of generated segments per class seen by LCSS algorithm for GAN and TS-SGAN for ARAT 0.

Segment	1	2	3
TS-SGAN	492	500	508

Table 5.9: The number of generated segments per class seen by LCSS algorithm for GAN and TS-SGAN for ARAT 1.

Segment	1	2	3	4	5	6
TS-SGAN	339	206	292	200	214	249

The tables clearly demonstrate that the TS-SGAN-generated data are uniformly distributed across various segments and classes of the dataset, which was not the case with the GAN-generated data. Furthermore, training the same classifier as in Section

Table 5.10: The number of generated segments per class seen by LCSS algorithm for GAN and TS-SGAN for ARAT 2.

Segment	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
TS-SGAN	57	33	22	47	65	73	25	40	45	50	25	32	45	28	50	54	31	57	10	44	19	28	60	10	85	64	21	29	29	35	74	32	50	30	35	6	35	25

Table 5.11: The number of generated segments per class seen by LCSS algorithm for GAN and TS-SGAN for ARAT 3.

Segment	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
TS-SGAN	69	52	26	73	35	65	61	43	12	118	14	65	50	13	44	16	26	20	24	45	127	30	87	43	62	16	107	53	11	74	19

5.3.3 on the TS-SGAN-generated data yields an accuracy of 98.2% and a weighted F1-score of 0.99, surpassing the GAN model's performance by 35% in accuracy and 0.41 in F1-score per class. These results indicate that the TS-SGAN model is not prone to mode collapse and effectively captures all relevant information while generating synthetic data for this dataset.

Activity recognition dataset

Initially, the SN was trained on the authentic dataset consisting of 14400 segments, for a total of 150 epochs, with the aim of acquiring the ability to distinguish between the various classes.

Subsequently, utilising the identical parameters from Section 5.3.3, the TS-SGAN was trained, and 90000 segments were generated. The class distribution per partition is illustrated in Figure 5.7. It is observed that akin to the findings obtained for the ARAT dataset, the TS-SGAN-generated data is uniformly distributed among the classes, as opposed to what was observed for the GAN-generated data.

Furthermore, upon training the previously established classifier on the generated data

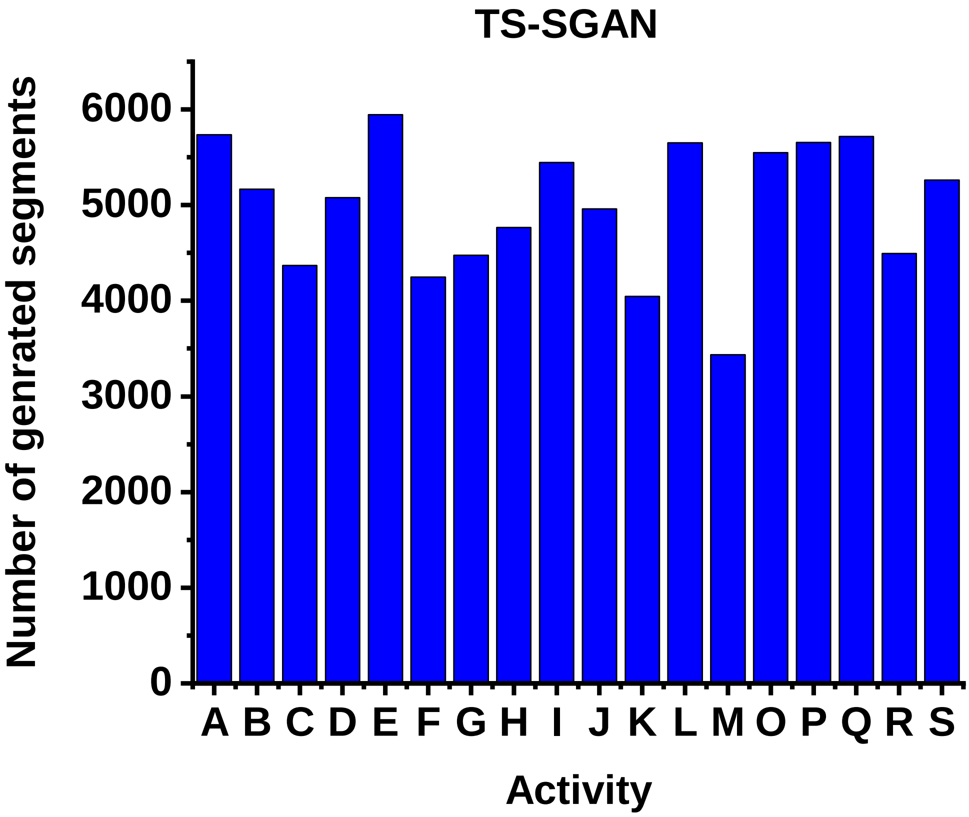


Figure 5.7: Distribution of the TS-SGAN generated data for different classes.

and then using the authentic data as a validation set, an accuracy of 90.8% with an F1-score per class of 0.90 was obtained. This represents a substantial improvement of 42.07% in accuracy and 0.45 in F1-score as compared to the GAN-generated data.

5.4.4 Results Summary

In this study, a framework for generating realistic and heterogeneous multivariate TS data is introduced. Our approach leverages two datasets related to post-stroke rehabilitation assessment: (1) a small, unbalanced dataset containing data for a popular rehabilitation assessment scale, and (2) a larger activity recognition dataset. data that closely resembled the original data and contained enough information to enable near-perfect classification by a classifier are generated.

Initially, a vanilla GAN is employed to generate the multivariate TS data. the model is trained separately on each category in the first dataset, as it is not feasible to predict the label from visual inspection of the signal. In the case of the second dataset, the model is trained on the entire dataset, as it was easier to distinguish between classes. While the generated data met the realism criteria, further inspection revealed that many segments from the dataset were not included in the generated data. This observation suggested that the model suffered from mode collapse, failing to capture the heterogeneity of the true distribution. To confirm this hypothesis, the LCSS algorithm is used to compute the similarity between the generated data and the original segments. Results on both datasets confirmed the presence of mode collapse. Additionally, a classifier is trained on the generated data and evaluated using the real data as a validation set, and the classification results were poor due to mode collapse.

To address the issue of mode collapse, a new method that uses the TS-SGAN is proposed. This involved adding a layer that learns the heterogeneity between different input structures and uses this information to characterise all modes, enabling the generator to spawn more diverse data. This was achieved by adding an SN that first discriminates between the dataset elements and then generates similarity, which is sent to a second discriminator in the network. The resulting generator did not suffer from mode collapse, and the generated data were heterogeneous, as evidenced by the distribution of the generated data across both datasets and the excellent classification results when training the classifier on the newly generated data.

This study lays the groundwork for future research endeavours. Our intention is to extend the application of the TS-SGAN model to other types of GAN architectures, such as conditional GANs and cycle GANs, and assess their efficacy on various TS datasets. While our focus has been on post-stroke rehabilitation, This model is believed to have the potential to be applied to other TS domains with further research and development. Additionally, conducting a thorough analysis of the generated data, including their variability, and performing comprehensive comparisons with the original datasets, could serve as valuable groundwork for future research.

5.5 Conclusion

The assessment of post-stroke telerehabilitation heavily relies on a substantial amount of data collected from WS. The accuracy of the patient modelling depends on the realism and diversity of this data, making data augmentation in the TS domain a critical step in the development of more efficient assessment models that can generalize to real-life circumstances. Despite their ability to generate meaningful data, GANs are known to suffer from mode collapse. To address this limitation, a new model, called TS-SGAN is proposed, which incorporates a second discriminator and SN. This modification effectively resolves the issue of mode collapse and enables the generation of more heterogeneous data, the resulting model was able to generate more diverse and realistic data, which improved the classification performance of the activity recognition dataset from 48.73% to 90.8% and from 63% to 98.2% for the ARAT dataset. This new model represents a solution to the data quantity limitation in post-stroke rehabilitation that was identified in Chapter 2 (data from post-stroke rehabilitation assessment is not very abundant due to the complexities involved in collecting data within the healthcare sector).

Chapter 6

Conclusion

In this comprehensive dissertation, an extensive and systematic exploration of the post-stroke telerehabilitation assessment field was undertaken, resulting in the formulation of a novel taxonomy that categorises the domain into three distinct areas: activity recognition, movement classification, and clinical assessment emulation. The in-depth analysis encompassed multiple facets, ranging from the prevalent use of IMU sensors to the emerging utilisation of EMG sensors in data collection, as well as scrutinising sensor placements, study designs, and the intricacies of feature engineering techniques. Notably, the landscape of machine learning in this domain has been evolving rapidly, with the advent of sophisticated algorithms, particularly deep learning, which have mitigated the requirement for extensive domain expertise.

Furthermore, this dissertation identified and addressed several pivotal challenges faced by researchers in the field. These challenges spanned issues related to data availability and quality, recruitment complexities, the complex real-world conditions under which assessments are conducted, concerns regarding the power consumption of wearable sensors, and the critical factor of patient acceptance in telerehabilitation systems. To facilitate the progress of fellow researchers, the dissertation offered valuable insights and practical tips to enhance the development of telerehabilitation systems.

The research endeavours also encompassed an in-depth investigation into the performance of cutting-edge ts dl algorithms, particularly on a complex HAR dataset. The study systematically examined the impact of sensor types, their specific placements on the human body, and the various data pre-processing techniques employed. The results unveiled that multiple algorithms yielded closely comparable accuracy rates, with Xceptiontime slightly outperforming others. Notably, the placement of sensors on the body emerged as a pivotal determinant in achieving precise activity recognition, with certain sensor placements displaying heightened sensitivity to specific activities. A noteworthy discovery was that hand sensors achieved a significantly higher accuracy rate (84%) compared to waist sensors (42%).

Moreover, this dissertation introduced an innovative pipeline for HAR assessment. It entailed the transformation of segmented time series data into images through diverse encoding techniques, including GASF, GADF, and MKV. These images, stemming from different axes of the sensors, were seamlessly fused to create multi-channel images. The application of linear interpolation was employed to augment data windows and subsequently enhance the sizes of the images. Comparative analysis revealed that increasing image size improved classification accuracy up to a specific threshold before diminishing returns set in. Conversely, a decrease in image size from the original window length struck an optimal balance between performance accuracy and computational complexity. The fusion of these multi-channel images notably boosted accuracy from 83% for one-dimensional models to a remarkable 91.5% when utilising GASF encoding with 300-size images, though MKV encoding exhibited comparatively lower performance.

Moreover, the dissertation unveiled an innovative solution to combat the mode collapse issue frequently encountered in conventional GANs. The novel approach introduced TS-SGAN, an advanced model that incorporated a secondary discriminator and an SN. This groundbreaking innovation led to the generation of a more diverse and realistic dataset, substantially elevating the classification performance for activity recognition datasets. This advancement is particularly evident in the performance improvements achieved in the ARAT dataset, where accuracy surged from a baseline of 63% to an impressive 98.2%. This innovative approach effectively addresses the challenge of limited data quantity, which was identified in Chapter 2, thereby significantly enhancing the efficacy of post-stroke rehabilitation assessment models with more authentic and varied datasets.

The contributions of this research are summarised below:

- Development of a New Taxonomy: A new taxonomy for post-stroke telerehabilitation assessment is proposed, categorising the field into activity recognition, movement classification, and clinical assessment emulation. This taxonomy provides a comprehensive framework for understanding and organising the different aspects of assessment in this domain.
- Evaluation of Existing Research: A thorough evaluation of numerous research works in post-stroke telerehabilitation assessment is conducted. This evaluation covers various aspects such as sensor usage, sensor placements, study designs, feature engineering, and machine learning techniques employed for assessment.
- Identification of Limitations and Challenges: Based on the reviewed literature, the study identifies common limitations and challenges in the field, including issues related to data volume and quality, recruitment difficulties, field complexity, power consumption, and patients' acceptance. These findings highlight areas where improvements and advancements are needed.
- Enhancement of Evaluation Algorithm: Enhancing the efficacy of the evaluation algorithm used in post-stroke rehabilitation assessment. this was done by adapting seventeen different CV models to Ts domain.
- Proposing a new pipeline to improve the accuracy of assessment, is done by encoding each axis of the sensor into an image and fusing all the images together to have a multi-channel image. Moreover, a linear interpolation was done on

the original TS data in order to control the size of the resulting images. This allowed to improve the accuracy of the classification of different datasets.

- Generation of Authentic and Informative Data: Another contribution focuses on generating more authentic and informative data for post-stroke rehabilitation assessment. This contribution involves the development of a novel algorithm or methodology to create realistic and heterogeneous datasets that can enhance the accuracy and effectiveness of assessment models. This was done by coupling a Siamese network with a GAN to add the heterogeneity aspect.

6.1 Future works

Some potential directions for this specific study include:

- A data collection study at Colchester Hospital, targeting post-stroke patients who are performing sit-to-stand rehabilitation exercises. Patients will be equipped with sensing devices, including a shimmer device with an IMU and non-invasive surface EMG sensors, to capture body movements and muscle activity. The goal is to use this data to develop a robust system for evaluating the effectiveness of rehabilitation exercises. Feedback from rehabilitation trainers will also be recorded and used as a reference in the assessment process. This research at East Suffolk & North Essex NHS Foundation Trust Colchester Hospital aims to gather valuable data to improve and refine assessment systems for post-stroke rehabilitation, enhancing the accuracy and reliability of evaluations for better patient outcomes.
- A promising research direction includes analysing the results obtained from processing sensor data through each layer of adapted CV models. This analysis aims to gain a deeper understanding of the outcomes at each layer to enhance the accuracy of assessment algorithms. By examining the intermediate outputs

of the CV models, researchers can identify areas for improvement and optimise the overall system performance.

- Another avenue for exploration is adapting NLP models to the time series sensor domain. This involves assessing the performance of NLP models when applied to the analysis of sequential sensor data, to understand the temporal patterns and inherent characteristics present. This exploration could provide valuable insights into using NLP techniques to improve the assessment of post-stroke rehabilitation, expanding the range of methods for accurate and comprehensive evaluation.

Some potential study directives and tips that might be worth considering to address a few challenges encountered in post-stroke rehabilitation in general:

- Provide a person-centric approach that considers both what the individual should and can achieve during rehabilitation. Indeed, integrating the quantification and analysis of the present and future conditions of the patient would result in a personalised treatment that takes into account the specificities of the different users.

- Employing additional sensors in conjunction with IMUs to model additional quantities to limb kinematics depending on the exercise. For batteries of tests that involve strength exercises, employing EMG sensors would be an interesting approach to have muscular activity, for gait-related tasks and sit/standing, using an insole pressure sensor would add useful information related to which lower limb is more active. For exercises that involve changing body level like standing up sitting down or going upstairs using level sensors would be an interesting approach. This will permit to lay out a more holistic and subjective assessment of the movement dysfunction

- Employing non-invasive, unobtrusive WS and taking into more consideration the patient's comfort as now many studies proved the possibility to design effective systems with a very small number of sensors (sometimes a single sensor is sufficient) as is the case in [62, 70]. Moreover, making the system simple to use providing visual tips as using avatars, and giving positive feedback on the execution would attract more users.

- Implementing a more holistic assessment system by combining multiple evaluation categories as a movement classification in conjunction with clinical assessment emulation, would allow for having different and complementary perspectives and therefore a more effective assessment. From the works reviewed, no paper combined it.
- Taking the assistance of field professionals when designing the systems, as some stroke clinical assessments, and some severe stroke cases require particular expertise in dealing with patients for example to position their limbs when doing their recovery tasks.
- Making more use of DL algorithms as they do not require thorough feature engineering and thus require less signal processing expertise. Moreover, 1D TS DL has emerged and they provide better accuracies than conventional ML algorithms and are even much faster like [129].
- As AI-based technologies are going to be an important part of the modern world, it is important to follow universal standards and guidelines that orient the patients' well-being in light of the social and ethical issues of these AI technologies. The recent IEEE 7010-2020 [183] is one good example.

References

[1] E. J. Benjamin, S. S. Virani, C. W. Callaway, A. M. Chamberlain, A. R. Chang, S. Cheng, S. E. Chiuve, M. Cushman, F. N. Delling, R. Deo *et al.*, "Heart disease and stroke statistics—2018 update: a report from the american heart association," *Circulation*, vol. 137, no. 12, pp. e67–e492, 2018.

[2] P. Langhorne, J. Bernhardt, and G. Kwakkel, "Stroke rehabilitation," *The Lancet*, vol. 377, no. 9778, pp. 1693–1702, 2011.

[3] E. H. Lo, M. A. Moskowitz, and T. P. Jacobs, "Exciting, radical, suicidal: how brain cells die after stroke," *Stroke*, vol. 36, no. 2, pp. 189–192, 2005.

[4] J. Gomes and A. M. Wachsman, "Types of strokes," in *Handbook of clinical nutrition and stroke.* Springer, 2013, pp. 15–31.

[5] S. C. Johnston, "Transient ischemic attack," *New England Journal of Medicine*, vol. 347, no. 21, pp. 1687–1692, 2002.

[6] H. B. Van der Worp and J. van Gijn, "Acute ischemic stroke," *New England Journal of Medicine*, vol. 357, no. 6, pp. 572–579, 2007.

[7] S. D. Smith and C. J. Eskey, "Hemorrhagic stroke," *Radiologic Clinics*, vol. 49, no. 1, pp. 27–45, 2011.

[8] A. Rosen, "Stroke: who's counting what?" *Development*, vol. 38, no. 2, pp. 281–289, 2001.

[9] J. Gomes and A. M. Wachsman, "Types of strokes," in *Handbook of clinical nutrition and stroke.* Totowa, New Jersey: Springer, 2013, pp. 15–31.

[10] P. Pound, P. Gompertz, and S. Ebrahim, “Illness in the context of older age: the case of stroke,” *Sociology of health & illness*, vol. 20, no. 4, pp. 489–506, 1998.

[11] D. E. Levy, R. L. Van Uitert, and L. Catherine, “Delayed postischemic hypoperfusion: a potentially damaging consequence of stroke,” *Neurology*, vol. 29, no. 9 Part 1, pp. 1245–1245, 1979.

[12] N. H. S. Constitution, “Rehabilitation after a strok,” *Proteins*, pp. 1–17, 2013.

[13] B. H. Dobkin, “Strategies for stroke rehabilitation,” *The Lancet Neurology*, vol. 3, no. 9, pp. 528–536, 2004.

[14] S. A. Billinger, R. Arena, J. Bernhardt, J. J. Eng, B. A. Franklin, C. M. Johnson, M. MacKay-Lyons, R. F. Macko, G. E. Mead, E. J. Roth *et al.*, “Physical activity and exercise recommendations for stroke survivors: a statement for healthcare professionals from the american heart association/american stroke association,” *Stroke*, vol. 45, no. 8, pp. 2532–2553, 2014.

[15] R. L. Sacco, R. Adams, G. Albers, M. J. Alberts, O. Benavente, K. Furie, L. B. Goldstein, P. Gorelick, J. Halperin, R. Harbaugh *et al.*, “Guidelines for prevention of stroke in patients with ischemic stroke or transient ischemic attack: a statement for healthcare professionals from the american heart association/american stroke association council on stroke: co-sponsored by the council on cardiovascular radiology and intervention: the american academy of neurology affirms the value of this guideline.” *Stroke*, vol. 37, no. 2, pp. 577–617, 2006.

[16] J. A. Zivin and D. W. Choi, “Stroke therapy,” *Scientific American*, vol. 265, no. 1, pp. 56–65, 1991.

[17] L. Catanese, J. Tarsia, and M. Fisher, “Acute ischemic stroke therapy overview,” *Circulation research*, vol. 120, no. 3, pp. 541–558, 2017.

[18] M. Fisher and W. Schaebitz, “An overview of acute stroke therapy: past, present, and future,” *Archives of internal medicine*, vol. 160, no. 21, pp. 3196–3206, 2000.

[19] J.-F. Nepveu, A. Thiel, A. Tang, J. Fung, J. Lundbye-Jensen, L. A. Boyd, and M. Roig, “A single bout of high-intensity interval training improves motor skill retention in individuals with stroke,” *Neurorehabilitation and neural repair*, vol. 31, no. 8, pp. 726–735, 2017.

[20] R. A. Geiger, J. B. Allen, J. O’Keefe, and R. R. Hicks, “Balance and mobility following stroke: effects of physical therapy interventions with and without biofeedback/forceplate training,” *Physical therapy*, vol. 81, no. 4, pp. 995–1005, 2001.

[21] E. Taub, J. E. Crago, and G. Uswatte, “Constraint-induced movement therapy: A new approach to treatment in physical rehabilitation.” *Rehabilitation Psychology*, vol. 43, no. 2, p. 152, 1998.

[22] Z.-S. Hosseini, H. Peyrovi, and M. Gohari, “The effect of early passive range of motion exercise on motor function of people with stroke: a randomized controlled trial,” *Journal of caring sciences*, vol. 8, no. 1, p. 39, 2019.

[23] J. R. Gladman and C. Sackley, “The scope for rehabilitation in severely disabled stroke patients,” *Disability and rehabilitation*, vol. 20, no. 10, pp. 391–394, 1998.

[24] C. M. Stinear, C. E. Lang, S. Zeiler, and W. D. Byblow, “Advances and challenges in stroke rehabilitation,” *The Lancet Neurology*, vol. 19, no. 4, pp. 348–360, 2020.

[25] J. A. Young and M. Tolentino, “Stroke evaluation and treatment,” *Topics in stroke rehabilitation*, vol. 16, no. 6, pp. 389–410, 2009.

[26] A. E. Dickerson, L. J. Molnar, D. W. Eby, G. Adler, M. Bedard, M. Berg-Weger, S. Classen, D. Foley, A. Horowitz, H. Kerschner *et al.*, “Transportation and aging:

A research agenda for advancing safe mobility," *The Gerontologist*, vol. 47, no. 5, pp. 578–590, 2007.

[27] J. H. Cauraugh and S. B. Kim, "Chronic stroke motor recovery: duration of active neuromuscular stimulation," *Journal of the neurological sciences*, vol. 215, no. 1-2, pp. 13–19, 2003.

[28] L. H. Schwamm, R. G. Holloway, P. Amarenco, H. J. Audebert, T. Bakas, N. R. Chumbler, R. Handschu, E. C. Jauch, W. A. Knight IV, S. R. Levine *et al.*, "A review of the evidence for the use of telemedicine within stroke systems of care: a scientific statement from the american heart association/american stroke association," *Stroke*, vol. 40, no. 7, pp. 2616–2634, 2009.

[29] T. Hester, R. Hughes, D. M. Sherrill, B. Knorr, M. Akay, J. Stein, and P. Bonato, "Using wearable sensors to measure motor abilities following stroke," in *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06).* IEEE, 2006, pp. 4–pp.

[30] Y. Zhou, Y. Fang, J. Zeng, K. Li, and H. Liu, "A multi-channel emg-driven fes solution for stroke rehabilitation," in *International Conference on Intelligent Robotics and Applications.* Springer, 2018, pp. 235–243.

[31] K. Rathakrishnan, S.-N. Min, and S. J. Park, "Evaluation of ecg features for the classification of post-stroke survivors with a diagnostic approach," *Applied Sciences*, vol. 11, no. 1, p. 192, 2021.

[32] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "gaftion," *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, p. 21, 2012.

[33] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (imu) sensor applications," *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.

[34] R. W. Bohannon, “Sit-to-stand test for measuring performance of lower extremity muscles,” *Perceptual and motor skills*, vol. 80, no. 1, pp. 163–166, 1995.

[35] D. Podsiadlo and S. Richardson, “The timed “up & go”: a test of basic functional mobility for frail elderly persons,” *Journal of the American geriatrics Society*, vol. 39, no. 2, pp. 142–148, 1991.

[36] S. Díaz, J. B. Stephenson, and M. A. Labrador, “Use of wearable sensor technology in gait, balance, and range of motion analysis,” *Applied Sciences*, vol. 10, no. 1, p. 234, 2020.

[37] K. Shailaja, B. Seetharamulu, and M. Jabbar, “Machine learning in healthcare: A review,” in *2018 Second international conference on electronics, communication and aerospace technology (ICECA).* IEEE, 2018, pp. 910–914.

[38] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, “Machine learning in wireless sensor networks: Algorithms, strategies, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[39] I. Boukhennoufa, X. Zhai, V. Utti, J. Jackson, and K. D. McDonald-Maier, “Wearable sensors and machine learning in post-stroke rehabilitation assessment: A systematic review,” *Biomedical Signal Processing and Control*, vol. 71, p. 103197, 2022.

[40] I. B. Abdallah and Y. Bouteraa, “An optimized stimulation control system for upper limb exoskeleton robot-assisted rehabilitation using a fuzzy logic-based pain detection approach,” *Sensors*, vol. 24, no. 4, p. 1047, 2024.

[41] X. Hu, K. Tong, X. Wei, W. Rong, E. Susanto, and S. Ho, “The effects of post-stroke upper-limb training with an electromyography (emg)-driven hand robot,” *Journal of Electromyography and Kinesiology*, vol. 23, no. 5, pp. 1065–1074, 2013.

[42] H. Yang, J. Wan, Y. Jin, X. Yu, and Y. Fang, “Eeg and emg driven post-stroke rehabilitation: A review,” *IEEE Sensors Journal*, 2022.

[43] ——, “Eeg and emg driven post-stroke rehabilitation: A review,” *IEEE Sensors Journal*, 2022.

[44] J. C. Pérez-Ibarra and A. A. Siqueira, “Comparison of kinematic and emg parameters between unassisted, fixed-and adaptive-stiffness robotic-assisted ankle movements in post-stroke subjects,” in *2017 International Conference on Rehabilitation Robotics (ICORR).* IEEE, 2017, pp. 461–466.

[45] ——, “Comparison of kinematic and emg parameters between unassisted, fixed-and adaptive-stiffness robotic-assisted ankle movements in post-stroke subjects,” in *2017 International Conference on Rehabilitation Robotics (ICORR).* IEEE, 2017, pp. 461–466.

[46] ——, “Comparison of kinematic and emg parameters between unassisted, fixed-and adaptive-stiffness robotic-assisted ankle movements in post-stroke subjects,” in *2017 International Conference on Rehabilitation Robotics (ICORR).* IEEE, 2017, pp. 461–466.

[47] O. E. Ogul, D. K. Coskunsu, S. Akcay, K. Akyol, L. Hanoglu, and N. Ozturk, “The effect of electromyography (emg)-driven robotic treatment on the recovery of the hand nine years after stroke,” *Journal of Hand Therapy*, vol. 36, no. 1, pp. 234–240, 2023.

[48] Y. Murakami, K. Honaga, H. Kono, K. Haruyama, T. Yamaguchi, M. Tani, R. Isayama, T. Takakura, A. Tanuma, K. Hatori *et al.*, “New artificial intelligence-integrated electromyography-driven robot hand for upper extremity rehabilitation of patients with stroke: A randomized, controlled trial,” *Neurorehabilitation and Neural Repair*, p. 15459683231166939, 2023.

[49] A. P. Arantes, N. Bressan, L. R. Borges, and C. A. McGibbon, "Evaluation of a novel real-time adaptive assist-as-needed controller for robot-assisted upper extremity rehabilitation following stroke," *Plos one*, vol. 18, no. 10, p. e0292627, 2023.

[50] I. B. Abdallah and Y. Bouteraa, "A newly-designed wearable robotic hand exoskeleton controlled by emg signals and ros embedded systems," *Robotics*, vol. 12, no. 4, p. 95, 2023.

[51] D. Su, Z. Hu, J. Wu, P. Shang, and Z. Luo, "Review of adaptive control for stroke lower limb exoskeleton rehabilitation robot based on motion intention recognition," *Frontiers in Neurorobotics*, vol. 17, 2023.

[52] D. Jack, R. Boian, A. S. Merians, M. Tremaine, G. C. Burdea, S. V. Adamovich, M. Recce, and H. Poizner, "Virtual reality-enhanced stroke rehabilitation," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 9, no. 3, pp. 308–318, 2001.

[53] K. E. Laver, B. Lange, S. George, J. E. Deutsch, G. Saposnik, and M. Crotty, "Virtual reality for stroke rehabilitation," *Cochrane database of systematic reviews*, no. 11, 2017.

[54] ——, "Virtual reality for stroke rehabilitation," *Stroke*, vol. 49, no. 4, pp. e160–e161, 2018.

[55] G. Saposnik, M. Levin, and S. O. R. C. S. W. Group, "Virtual reality in stroke rehabilitation: a meta-analysis and implications for clinicians," *Stroke*, vol. 42, no. 5, pp. 1380–1386, 2011.

[56] A. Demeco, L. Zola, A. Frizziero, C. Martini, A. Palumbo, R. Foresti, G. Buccino, and C. Costantino, "Immersive virtual reality in post-stroke rehabilitation: a systematic review," *Sensors*, vol. 23, no. 3, p. 1712, 2023.

[57] N. Aderinto, G. Olatunji, M. O. Abdulbasit, M. Edun, G. Aboderin, and E. Egbunu, "Exploring the efficacy of virtual reality-based rehabilitation in stroke: a narrative review of current evidence," *Annals of Medicine*, vol. 55, no. 2, p. 2285907, 2023.

[58] J. Hao, Z. Yao, K. Harp, D. Y. Gwon, Z. Chen, and K.-C. Siu, "Effects of virtual reality in the early-stage stroke rehabilitation: A systematic review and meta-analysis of randomized controlled trials," *Physiotherapy Theory and Practice*, vol. 39, no. 12, pp. 2569–2588, 2023.

[59] J. Hao, Z. He, X. Yu, and A. Remis, "Comparison of immersive and non-immersive virtual reality for upper extremity functional recovery in patients with stroke: A systematic review and network meta-analysis," *Neurological Sciences*, pp. 1–19, 2023.

[60] G. Yang, J. Deng, G. Pang, H. Zhang, J. Li, B. Deng, Z. Pang, J. Xu, M. Jiang, P. Liljeberg *et al.*, "An iot-enabled stroke rehabilitation system based on smart wearable armband and machine learning," *IEEE journal of translational engineering in health and medicine*, vol. 6, pp. 1–10, 2018.

[61] M. Bobin, H. Amroun, M. Boukalle, M. Anastassova, and M. Ammi, "Smart cup to monitor stroke patients activities during everyday life," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 189–195.

[62] M. Panwar, D. Biswas, H. Bajaj, M. Jöbges, R. Turk, K. Maharatna, and A. Acharyya, "Rehab-net: Deep learning framework for arm movement classification using wearable sensors for stroke rehabilitation," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 11, pp. 3026–3037, 2019.

[63] N. A. Capela, E. D. Lemaire, and N. Baddour, "Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients," *PloS one*, vol. 10, no. 4, p. e0124414, 2015.

[64] S. H. Chae, Y. Kim, K.-S. Lee, and H.-S. Park, "Development and clinical evaluation of a web-based upper limb home rehabilitation system using a smartwatch and machine learning model for chronic stroke survivors: Prospective comparative study," *JMIR mHealth and uHealth*, vol. 8, no. 7, p. e17216, 2020.

[65] T. Tran, L.-C. Chang, I. Almubark, E. M. Bochniewicz, L. Shu, P. S. Lum, and A. Dromerick, "Robust classification of functional and nonfunctional arm movement after stroke using a single wrist-worn sensor device," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 5457–5459.

[66] M. K. O'Brien, N. Shawen, C. K. Mummidisetty, S. Kaur, X. Bo, C. Poellabauer, K. Kording, and A. Jayaraman, "Activity recognition for persons with stroke using mobile phone technology: toward improved performance in a home setting," *Journal of medical Internet research*, vol. 19, no. 5, p. e184, 2017.

[67] F. Massé, R. R. Gonzenbach, A. Arami, A. Paraschiv-Ionescu, A. R. Luft, and K. Aminian, "Improving activity recognition using a wearable barometric pressure sensor in mobility-impaired stroke patients," *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, pp. 1–15, 2015.

[68] P.-W. Chen, N. A. Baune, I. Zwir, J. Wang, V. Swamidass, and A. W. Wong, "Measuring activities of daily living in stroke patients with motion machine learning algorithms: A pilot study," *International Journal of Environmental Research and Public Health*, vol. 18, no. 4, p. 1634, 2021.

[69] I. Boukhennoufa, X. Zhai, K. D. McDonald-Maier, V. Utti, and J. Jackson, "Improving the activity recognition using gmaf and transfer learning in post-stroke

rehabilitation assessment," in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2021, pp. 000 391–000 398.

[70] L. Meng, A. Zhang, C. Chen, X. Wang, X. Jiang, L. Tao, J. Fan, X. Wu, C. Dai, Y. Zhang *et al.*, "Exploration of human activity recognition using a single sensor for stroke survivors and able-bodied people," *Sensors*, vol. 21, no. 3, p. 799, 2021.

[71] A. David, R. Ramadoss, A. Ramachandran, and S. Sivapatham, "Activity recognition of stroke-affected people using wearable sensor," *ETRI Journal*, vol. 21, no. 3, 2023.

[72] S. I. Lee, C. P. Adans-Dester, M. Grimaldi, A. V. Dowling, P. C. Horak, R. M. Black-Schaffer, P. Bonato, and J. T. Gwin, "Enabling stroke rehabilitation in home and community settings: a wearable sensor-based approach for upper-limb motor training," *IEEE journal of translational engineering in health and medicine*, vol. 6, pp. 1–11, 2018.

[73] X. Liu, S. Rajan, N. Ramasarma, P. Bonato, and S. I. Lee, "The use of a finger-worn accelerometer for monitoring of hand use in ambulatory settings," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 599–606, 2019.

[74] A. Kaku, A. Parnandi, A. Venkatesan, N. Pandit, H. Schambra, and C. Fernandez-Granda, "Towards data-driven stroke rehabilitation via wearable sensors and deep learning," *arXiv preprint arXiv:2004.08297*, vol. 126, 2020.

[75] I. Bisio, C. Garibotto, F. Lavagetto, and A. Sciarrone, "When ehealth meets iot: A smart wireless system for post-stroke home rehabilitation," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 24–29, 2019.

[76] A. Mannini, D. Trojaniello, A. Cereatti, and A. M. Sabatini, "A machine learning framework for gait classification using inertial sensors: Application to elderly,

post-stroke and huntington's disease patients," *Sensors*, vol. 16, no. 1, p. 134, 2016.

[77] A. H. Butt, C. Zambrana, S. Idelsohn-Zielonka, M. Claramunt-Molet, A. Ugartemendia-Etxarri, E. Rovini, A. Moschetti, C. Molleja, C. Martin, E. O. Salleras *et al.*, "Assessment of purposeful movements for post-stroke patients in activites of daily living with wearable sensor device," in *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB).* IEEE, 2019, pp. 1–8.

[78] A. Miller, L. Quinn, S. V. Duff, and E. Wade, "Comparison of machine learning approaches for classifying upper extremity tasks in individuals post-stroke," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).* IEEE, 2020, pp. 4330–4336.

[79] W.-C. Hsu, T. Sugiarto, Y.-J. Lin, F.-C. Yang, Z.-Y. Lin, C.-T. Sun, C.-L. Hsu, and K.-N. Chou, "Multiple-wearable-sensor-based gait classification and analysis in patients with neurological disorders," *Sensors*, vol. 18, no. 10, p. 3397, 2018.

[80] F.-C. Wang, S.-F. Chen, C.-H. Lin, C.-J. Shih, A.-C. Lin, W. Yuan, Y.-C. Li, and T.-Y. Kuo, "Detection and classification of stroke gaits by deep neural networks employing inertial measurement units," *Sensors*, vol. 21, no. 5, p. 1864, 2021.

[81] Y. Jiang, Y. Qin, I. Kim, and Y. Wang, "Towards an iot-based upper limb rehabilitation assessment system," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* IEEE, 2017, pp. 2414–2417.

[82] A. Derungs, C. Schuster-Amft, and O. Amft, "Wearable motion sensors and digital biomarkers in stroke rehabilitation," *Current Directions in Biomedical Engineering*, vol. 6, no. 3, pp. 229–232, 2020.

[83] N. Balestra, G. Sharma, L. M. Riek, and A. Busza, "Automatic identification of upper extremity rehabilitation exercise type and dose using body-worn sensors and machine learning: A pilot study," *Digital Biomarkers*, vol. 5, no. 2, pp. 158–166, 2021.

[84] S. Sapienza, C. Adans-Dester, O. Anne, G. Vergara-Diaz, S. Lee, S. Patel, R. Black-Schaffer, R. Zafonte, P. Bonato, C. Meagher *et al.*, "Using a minimum set of wearable sensors to assess quality of movement in stroke survivors," in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE).* IEEE, 2017, pp. 284–285.

[85] E. M. Bochniewicz, G. Emmer, A. McLeod, J. Barth, A. W. Dromerick, and P. Lum, "Measuring functional arm movement after stroke using a single wrist-worn sensor and machine learning," *Journal of Stroke and Cerebrovascular Diseases*, vol. 26, no. 12, pp. 2880–2887, 2017.

[86] C. Adans-Dester, N. Hankov, A. O'Brien, G. Vergara-Diaz, R. Black-Schaffer, R. Zafonte, J. Dy, S. I. Lee, and P. Bonato, "Enabling precision rehabilitation interventions using wearable sensors and machine learning to track motor recovery," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–10, 2020.

[87] L. Yu, D. Xiong, L. Guo, and J. Wang, "A remote quantitative fugl-meyer assessment framework for stroke patients based on wearable sensor networks," *Computer methods and programs in biomedicine*, vol. 128, pp. 100–110, 2016.

[88] S. Chaeibakhsh, E. Phillips, A. Buchanan, and E. Wade, "Upper extremity post-stroke motion quality estimation with decision trees and bagging forests," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* IEEE, 2016, pp. 4585–4588.

[89] A. Lucas, J. Hermiz, J. Labuzetta, Y. Arabadzhi, N. Karanjia, and V. Gilja, "Use of accelerometry for long term monitoring of stroke patients," *IEEE journal*

of translational engineering in health and medicine, vol. 7, pp. 1–10, 2019.

[90] X. Chen, Y. Guan, J.-Q. Shi, X.-L. Du, and J. Eyre, "Automated stroke rehabilitation assessment using wearable accelerometers in free-living environments," *arXiv preprint arXiv:2009.08798*, vol. 7, 2020.

[91] T. K. Lee, K.-H. Leo, S. Sanei, E. Chew, and L. Zhao, "Triaxial rehabilitative data analysis incorporating matching pursuit," in *2017 25th European Signal Processing Conference (EUSIPCO).* IEEE, 2017, pp. 434–438.

[92] B. Oubre, J.-F. Daneault, H.-T. Jung, K. Whritenour, J. G. V. Miranda, J. Park, T. Ryu, Y. Kim, and S. I. Lee, "Estimating upper-limb impairment level in stroke survivors using wearable inertial sensors and a minimally-burdensome motor task," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 3, pp. 601–611, 2020.

[93] E. Park, K. Lee, T. Han, and H. S. Nam, "Automatic grading of stroke symptoms for rapid assessment using optimized machine learning and 4-limb kinematics: clinical validation study," *Journal of Medical Internet Research*, vol. 22, no. 9, p. e20641, 2020.

[94] C. Werner, J. G. Schönhammer, M. K. Steitz, O. Lambercy, A. R. Luft, L. Demkó, and C. A. Easthope, "Using wearable inertial sensors to estimate clinical scores of upper limb movement quality in stroke," *Frontiers in Physiology*, vol. 13, p. 877563, 2022.

[95] L. Guo, B. Zhang, J. Wang, Q. Wu, X. Li, L. Zhou, and D. Xiong, "Wearable intelligent machine learning rehabilitation assessment for stroke patients compared with clinician assessment," *Journal of Clinical Medicine*, vol. 11, no. 24, p. 7467, 2022.

[96] S. Katz, "Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living," *Journal of the American Geriatrics*

Society, vol. 31, no. 12, pp. 721–727, 1983.

[97] K. J. Sullivan, J. K. Tilson, S. Y. Cen, D. K. Rose, J. Hershberg, A. Correa, J. Gallichio, M. McLeod, C. Moore, S. S. Wu *et al.*, "Fugl-meyer assessment of sensorimotor function after stroke: standardized training procedure for clinical practice and clinical trials," *Stroke*, vol. 42, no. 2, pp. 427–432, 2011.

[98] Y.-W. Hsieh, I.-P. Hsueh, Y.-T. Chou, C.-F. Sheu, C.-L. Hsieh, and G. Kwakkel, "Development and validation of a short form of the fugl-meyer motor scale in patients with stroke," *Stroke*, vol. 38, no. 11, pp. 3052–3054, 2007.

[99] S. L. Wolf, J. P. McJunkin, M. L. Swanson, and P. S. Weiss, "Pilot normative database for the wolf motor function test," *Archives of physical medicine and rehabilitation*, vol. 87, no. 3, pp. 443–445, 2006.

[100] S. L. Wolf, P. A. Catlin, M. Ellis, A. L. Archer, B. Morgan, and A. Piacentino, "Assessing wolf motor function test as outcome measure for research in patients after stroke," *Stroke*, vol. 32, no. 7, pp. 1635–1639, 2001.

[101] M. McDonnell, "Action research arm test," *Aust J Physiother*, vol. 54, no. 3, p. 220, 2008.

[102] T. Da Roza, T. Mascarenhas, M. Araujo, V. Trindade, and R. N. Jorge, "Oxford grading scale vs manometer for assessment of pelvic floor strength in nulliparous sports students," *Physiotherapy*, vol. 99, no. 3, pp. 207–211, 2013.

[103] S. R. Barreca, P. W. Stratford, L. M. Masters, C. L. Lambert, J. Griffiths, and C. McBay, "Validation of three shortened versions of the chedoke arm and hand activity inventory," *Physiotherapy Canada*, vol. 58, no. 2, pp. 148–156, 2006.

[104] S. Majumder, T. Mondal, and M. J. Deen, "Wearable sensors for remote health monitoring," *Sensors*, vol. 17, no. 1, p. 130, 2017.

[105] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, “Reviews on various inertial measurement unit (imu) sensor applications,” *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.

[106] K. Tanja-Dijkstra and M. E. Pieterse, “The psychological effects of the physical healthcare environment on healthcare personnel,” *Cochrane database of systematic reviews*, no. 1, 2011.

[107] D. W. Nicoll, “Users as currency: Technology and marketing trials as naturalistic environments,” *The Information Society*, vol. 16, no. 4, pp. 303–310, 2000.

[108] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists.* " O'Reilly Media, Inc.", 2018.

[109] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: review, opportunities and challenges,” *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[110] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmaki, and H. T. Toivonen, “Time series segmentation for context recognition in mobile devices,” in *Proceedings 2001 IEEE International Conference on Data Mining.* IEEE, 2001, pp. 203–210.

[111] T.-c. Fu, “A review on time series data mining,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[112] J. Heaton, “An empirical analysis of feature engineering for predictive modeling,” in *SoutheastCon 2016.* IEEE, 2016, pp. 1–6.

[113] Z. Wang and T. Oates, “Imaging time-series to improve classification and imputation,” in *IJCAI'15: Proceedings of the 24th International Conference on Artificial Intelligence.* ACM, 2015, p. 3939–3945.

[114] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, “Artificial intelligence in healthcare: past, present and future,”

Stroke and vascular neurology, vol. 2, no. 4, 2017.

[115] T. J. Quinn, P. Langhorne, and D. J. Stott, "Barthel index for stroke trials: development, properties, and application," *Stroke*, vol. 42, no. 4, pp. 1146–1151, 2011.

[116] P. Khera and N. Kumar, "Role of machine learning in gait analysis: a review," *Journal of Medical Engineering & Technology*, vol. 44, no. 8, pp. 441–467, 2020.

[117] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[118] W. Johnson, O. Onuma, M. Owolabi, and S. Sachdev, "Stroke: a global response is needed," *Bulletin of the World Health Organization*, vol. 94, no. 9, p. 634, 2016.

[119] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," *Frontiers in Robotics and AI*, vol. 2, p. 28, 2015.

[120] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[121] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[122] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[123] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[124] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[125] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[126] G. Weiss, "WISDM Smartphone and Smartwatch Activity and Biometrics Dataset ," UCI Machine Learning Repository, 2019, DOI: https://doi.org/10.24432/C5HK59.

[127] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN).* IEEE, 2017, pp. 1578–1585.

[128] L. Rosafalco, A. Manzoni, S. Mariani, and A. Corigliano, "Fully convolutional networks for structural health monitoring through multivariate time series classification," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, no. 1, pp. 1–31, 2020.

[129] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[130] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, "Xceptiontime: independent time-window xceptiontime architecture for hand gesture classification," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 1304–1308.

[131] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[132] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 558–567.

[133] W. Tang, G. Long, L. Liu, T. Zhou, J. Jiang, and M. Blumenstein, "Rethinking 1d-cnn for time series classification: A stronger baseline," *arXiv preprint arXiv:2002.10061*, 2020.

[134] J. Wang, Z. Wang, J. Li, and J. Wu, "Multilevel wavelet decomposition network for interpretable time series analysis," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2437–2446.

[135] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. SPIE, 2019, pp. 369–386.

[136] C. Fukuchi, R. Fukuchi, and M. Duarte, "A public dataset of overground and treadmill walking kinematics and kinetics in healthy individuals," *PeerJ*, vol. 6, p. 4640, 2018.

[137] B. Liew, S. Morris, and K. Netto, "Defining gait patterns using parallel factor 2 (parafac2): A new analysis of previously published data," *Journal of Biomechanics*, vol. 90, p. 133–137, 2019.

[138] A. Schache and R. Baker, "On the expression of joint moments during gait," *Gait and Posture*, vol. 25, p. 440–452, 2007.

[139] M. Dwarampudi and N. Reddy, “Effects of padding on lstms and cnns,” 2019.

[140] M. Pogson, J. Verheul, M. Robinson, J. Vanrenterghem, and P. Lisboa, “A neural network method to predict taskand step-specific ground reaction force magnitudes from trunk accelerations during running activities,” *Medical Engineering & Physics*, vol. 78, p. 82–89, 2020.

[141] J. Burdack, F. Horst, S. Giesselbach, I. Hassan, S. Daffner, and W. Schöllhorn, 2020.

[142] F. Wouda, M. Giuberti, G. Bellusci, E. Maartens, J. Reenalda, B.-J. Beijnum, and P. Veltink, “Estimation of vertical ground reaction forces and sagittal knee kinematics during running using three inertial sensors,” *Frontiers in Physiology*, vol. 9, 2018.

[143] S. Indolia, A. Goswami, S. Mishra, and P. Asopa, “Conceptual understanding of convolutional neural network-a deep learning approach,” *Procedia Computer Science*, vol. 132, p. 679–688, 2018.

[144] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feed-forward neural networks",” 2010, proceedings of Machine Learning Research: PMLR).

[145] H. Dau, A. Bagnall, K. Kamgar, C.-C. Yeh, Y. Zhu, S. Gharghabi, C. Ratanamahatana, and E. Keogh, “The ucr time series archive,” *IEEE/CAA Journal of Automatica Sinica*, vol. 6, p. 1293–1305, 2019.

[146] B. Stetter, F. Krafft, S. Ringhof, T. Stein, and S. Sell, “A machine learning and wearable sensor based approach to estimate external knee flexion and adduction moments during various locomotion tasks,” *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 9, 2020.

[147] M. Boswell, S. Uhlrich, k. Kidziński, K. Thomas, J. Kolesar, G. Gold, G. Beaupre,

and S. Delp, "A neural network to predict the knee adduction moment in patients with osteoarthritis using anatomical landmarks obtainable from 2d video analysis," *Osteoarthritis and Cartilage*, vol. 29, p. 346–356, 2021.

[148] B. Liew, D. Rügamer, X. Zhai, Y. Wang, S. Morris, and K. Netto, "Comparing shallow, deep, and transfer learning in predicting joint moments in running," *Journal of Biomechanics*, vol. 129, p. 110820, 2021.

[149] V. M. Souza, D. F. Silva, and G. E. Batista, "Extracting texture features for time series classification," in *2014 22nd International Conference on Pattern Recognition.* IEEE, 2014, pp. 1425–1430.

[150] J. R. Paulo, G. Pires, and U. J. Nunes, "Cross-subject zero calibration driver's drowsiness detection: Exploring spatiotemporal image encoding of eeg signals for convolutional neural network classification," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 29, pp. 905–915, 2021.

[151] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[152] Z. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Workshops at the twenty-ninth AAAI conference on artificial intelligence*, vol. 1, 2015.

[153] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[154] S. Benavidez and D. McCreight, "A deep learning approach for human activity recognition project category: Other (time-series classification)," 2019.

[155] S. J. Saleh, S. Q. Ali, and A. M. Zeki, "Random forest vs. svm vs. knn in

classifying smartphone and smartwatch sensor data using crisp-dm," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI).* IEEE, 2020, pp. 1–4.

[156] K. Giorgas and I. Varlamis, "Online federated learning with imbalanced class distribution," in *24th Pan-Hellenic Conference on Informatics*, 2020, pp. 91–95.

[157] B. Oluwalade, S. Neela, J. Wawira, T. Adejumo, and S. Purkayastha, "Human activity recognition using deep learning models on smartphones and smartwatches sensor data," *arXiv preprint arXiv:2103.03836*, 2021.

[158] M. S. H. Bhuiyan, N. S. Patwary, P. K. Saha, and M. T. Hossain, "Sensor-based human activity recognition: A comparative study of machine learning techniques," in *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT).* IEEE, 2020, pp. 286–290.

[159] D. R. Brillinger, *Time series: data analysis and theory.* SIAM, 2001.

[160] D. Kairy, P. Lehoux, C. Vincent, and M. Visintin, "A systematic review of clinical outcomes, clinical process, healthcare utilization and costs associated with telerehabilitation," *Disability and rehabilitation*, vol. 31, no. 6, pp. 427–447, 2009.

[161] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, "Recent advances of generative adversarial networks in computer vision," *IEEE Access*, vol. 7, pp. 14 985–15 006, 2018.

[162] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[163] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, and S. Gómez-Canaval, "Data augmentation techniques in time series domain: a survey and taxonomy," *Neural Computing and Applications*, vol. 35, no. 14, pp. 10 123–10 145, 2023.

[164] D. Carroll, “A quantitative test of upper extremity function,” *Journal of chronic diseases*, vol. 18, no. 5, pp. 479–491, 1965.

[165] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[166] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[167] J. Yoon, D. Jarrett, and M. Van der Schaar, “Time-series generative adversarial networks,” *Advances in neural information processing systems*, vol. 32, 2019.

[168] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[169] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, “Indexing multi-dimensional time-series with support for multiple distance measures,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 216–225.

[170] I. Boukhennoufa, X. Zhai, V. Utti, J. Jackson, and K. D. McDonald-Maier, “Encoding sensors’ data into images to improve the activity recognition in post stroke rehabilitation assessment,” in *International Conference on Pattern Recognition and Artificial Intelligence*. Springer, 2022, pp. 114–123.

[171] ——, “A comprehensive evaluation of state-of-the-art time-series deep learning models for activity-recognition in post-stroke rehabilitation assessment,” in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 2242–2247.

[172] I. Boukhennoufa, X. Zhai, K. D. McDonald-Maier, V. Utti, and J. Jackson, "Improving the activity recognition using gmaf and transfer learning in post-stroke rehabilitation assessment," in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2021, pp. 000 391–000 398.

[173] G. Das, D. Gunopulos, and H. Mannila, "Finding similar time series," in *European Symposium on Principles of Data Mining and Knowledge Discovery*. Springer, 1997, pp. 88–100.

[174] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings 18th international conference on data engineering*. IEEE, 2002, pp. 673–684.

[175] S. Sanei, D. Jarchi, and A. G. Constantinides, "Body sensor networking, design and algorithms," in *John Wiley and Sons*, 2020, pp. 216–225.

[176] I. Boukhennoufa, Z. Altai, X. Zhai, V. Utti, K. D. McDonald-Maier, and B. X. Liew, "Predicting the internal knee abduction impulse during walking using deep learning," *Frontiers in Bioengineering and Biotechnology*, vol. 10, 2022.

[177] M. Allahyani, R. Alsulami, T. Alwafi, T. Alafif, H. Ammaer, S. Sabban, and X. Chen, "Sd2gan: A siamese dual discriminator generative adversarial network for mode collapse reduction," *vol*, vol. 1, pp. 1–9, 2021.

[178] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, 1993.

[179] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[180] V. Kumar BG, G. Carneiro, and I. Reid, "Learning local image descriptors

with deep siamese and triplet convolutional networks by minimising global loss functions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5385–5394.

[181] X. Ning, K. Gong, W. Li, L. Zhang, X. Bai, and S. Tian, "Feature refinement and filter network for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 9, pp. 3391–3402, 2020.

[182] R. B. Myerson, "Nash equilibrium and the history of economic theory," *Journal of Economic Literature*, vol. 37, no. 3, pp. 1067–1082, 1999.

[183] D. Schiff, A. Ayesh, L. Musikanski, and J. C. Havens, "Ieee 7010: A new standard for assessing the well-being implications of artificial intelligence," in *2020 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2020, pp. 2746–2753.

[184] G. Rilling, P. Flandrin, P. Goncalves *et al.*, "On empirical mode decomposition and its algorithms," in *IEEE-EURASIP workshop on nonlinear signal and image processing*, vol. 3, no. 3. NSIP-03, Grado (I), 2003, pp. 8–11.

[185] H. J. Michielsen, J. De Vries, and G. L. Van Heck, "Psychometric qualities of a brief self-rated fatigue measure: The fatigue assessment scale," *Journal of psychosomatic research*, vol. 54, no. 4, pp. 345–352, 2003.

Appendix A

Appendix

The below tables contain some useful information identified in interesting research papers in the field of post-stroke rehabilitation.

Study characteristics related to the WS used and its placement, the monitored exercises, the participants, the selected features and the ML algorithm used and the classification performance for the included papers are presented in table A.1. While Table A.2 presents a summary of each paper and the corresponding limitations and objectives.

Table A.1: Characteristics of the reviewed papers

Paper	Sensor/limb	Exercise	Participants	Feature	ML method	Best performance
				Activity recognition		
[60]	EMG/ Forearm	9 different hand gestures	3 AB [1]	Time domain features corresponding to variance, waveform length, root mean square, zero-crossing and auto-regressive coefficients. PCA was then applied to the resulting 56 features vector and the three first components that contributed with 95.86% of the overall information were utilised.	MLP and SVM	Accuracy of 96.25%
[61]	IMU, level sensor/ Hand	ADLs (Walking, standing, sitting, up/down, drinking)	15 AB	Discrete cosine transform on the segmented time series data signals to extract frequency domain features and regroup the energy in the low frequency coefficients.	SVM, MLP	Accuracy of more than 92% for SVM
[62]	Accelerometer/ Arm	ADLs (20 arm movements)	10 SP [2]	The data was segmented to time windows and down sampled and the normalised magnitude of the acceleration was used. The different segments are then labeled according to the activity. Two different configurations were used for the participants: naturalistic data where patients are in their houses and 97.89% on semi-naturalistic data where patients are in labs	CNN	Accuracies of 88.87% on the naturalistic data and 97.89% on the semi-naturalistic data respectively.
[63]	IMU/ Right-front hip	ADLs (41 mobility tasks)	15 AB, 17 El [3], 12 SP	Extracted a number of 76 time series features, relief-F, correlation-based feature selection and fast correlation based filter were then used to select the most relevant features.	Bayes, SVM and RT	Variant for different tasks

[1] AB: Able-Bodied

[2] SP: Stroke patients

[3] El: Elderly

[64]	IMU/ Hand and hips	Bilateral shoulder flexion with both hands interlocked; wall push exercise; active scapular exercise; and towel slide exercise.	23 SP	Raw data from gyroscope, accelerometer and the combination of both to Recognise and record the type and frequency of the rehabilitation exercises.	CNN	99.9%
[65]	IMU/ Wrist	ADLs (Doing the laundry, performing kitchen tasks, shopping related tasks, and making the bed.)	10 AB, 10 SP	Extracted overall and axial means, overall and axial variances, entropy, minima, and maxima. The feature vectors were then compressed using PCA to reduce the high from 11 to 3 columns.	K-Means, KNN, RF, SVM, RBF SVM	RF accuracy 83%
[66]	IMU, barometer/ Waist	ADLs (Sitting, Lying, Standing, Stairs Up, Stairs Down, and Walking)	30SP	Features included statistical measures of the sensor signal, its derivatives, and the frequency domain (mean, range,skewness...etc)	RF	Trained on stroke activity achieved 75%
[67]	IMU, barometer/ Sternum	ADLs (sitting, standing, walking, lying,sit-to-stand, stand-to-sit,walking up and down the stairs, taking the elevator, washing hands, eating, pouring and drinking water, sleeping, shoe lacing, reading the newspaper..etc)	12SP	Different algorithms developed from previous researches by detecting transitional phases for different ADLs	Hierarchical Fuzzy Inference System	70.3%
[68]	IMU/ Wrist, arm	ADLs (Chopping food, vacuuming, sweeping, spreading jam or butter, folding laundry, eating, brushing teeth...etc)	11SP	Time series features (mean, standard deviation, autocorrelation, and slope) and frequency domain features (not mentioned)	DT, RF, SVM, and eXtreme Gradient Boosting (XGBoost)	82%
[69]	IMU/ Waist	ADLs (Walking, walking up, walking down, sit to stand, stand to sit, laying)	30AB	Segmented data were encoded into images using GMAF technique	Different CNN models	VGG16 98.53%

[70]	IMU, sEMG/ Wrist, arms forearms, legs, ankles	ADLs (Walking, tooth brush, gace washing, drinking)	9SP, 14AB	Noise was filtered through Butterworth filter and band-pass, then data were normalised, then 28 different time-series features were extracted RMS, meqn, variance...etc	SVM (linear and rbf), Adaboost, KNN, RF, DT, KNN	SVM-rbf 82.47% .
[71]	IMUs/ forearms	Forward and Upward stretch, Fling, Hand-to-mouth, Swipe, Pouring	28AB	RMS, mean, variance, skewness...etc	LSTM	89% .
				Movement classification		
[72]	IMU/ Wrist	Motor tasks associated with the FMA	20 SP, 10 El	Applied the minimal-redundancy maximal-relevance algorithm on the minimum, maximum, range, mean, standard deviation, RMS values, and the number of zero crossings of the time-series data.	LR, RF	87% and 84.3% successively
[73]	Accelerometer/ Finger and wrist	Estimate amount of hand use	18 AB	Extracted multiple time-series features (mean, inter-quartile range, minimum and maximum, root mean square of the acceleration time-series, standard deviation, ratio of the energy at the dominant frequency to the entire signal, energy of the time-series, skewness, kurtosis, and signal entropy) then a correlation based feature selection was utilised to identify the most relevant features.	SVR	0.11 RMSE
[74]	IMU/ Arms and chest	ADLs (washing the face, applying deodorant, combing the hair, donning and doing glasses, preparing and eating a slice of bread, ...etc)	48 SP	Raw data to measure functional primitive	CNN	70 %
[75]	IMU and pressure sensors/ Legs and feet	extension and abduction of the legs, sit-to-stand, gait and Bipodaal Bridge	NA	Extracted 64 features consisting of mean and the variance for the different sensing nodes	TB, RT, hyper-plane , MLP	MLP reported the best F-measure with 97.9%

[76]	IMU/ Shanks	Gait	15 SP	Used a total of 18 features consisting of Hidden Markov Model (Log-likelihood, EL model, Log-likelihood PS model, Log-likelihood, HD model, Difference between log-likelihoods given EL and PS models, Difference between log-likelihoods given EL and HD models, Difference between log-likelihoods given PS and HD models) time (Mean value Evaluated, Standard deviation, Variance, Maximum, Minimum, Range) and frequency domain features (Power at first dominant frequency (P1), Power at second dominant frequency, First dominant frequency, Second dominant frequency, Total power (PT) , P1/PT).	SVM	LOSO cross validation and an accuracy of 90.5%
[77]	IMU/ index and finger	9 different ADLs: resting, eating, pouring water, drinking, brushing,folding towel, grasp bottle, grasp brush, and grasp towel.	10AB, 12SP	A low-pass fourth-order Butterworth filter was applied to all the signals to remove the tremor noise. A high pass fourth-order Butterworth filter was implemented for frequency analysis to eliminate the continuous component of the signal. then data was normalised then different features were extracted: skewness, average, RMS, jerk...etc	SVM, ANN	ANN 99.9% for a dataset containing both SP and AB.
[78]	IMU/ Wrist, arm, sternum	Uni-manual tasks, bi-manual asymmetric tasks, bi-manual symmetric tasks all performed with dominant and non-dominant hand	20SP, 20AB	Classifier Attribute Evaluator, ReliefF, Info Gain Attribute Evaluator and Gain Ratio Attribute were used to select the most relevant features then Root Mean Square, Mean, Signal Magnitude Area, Signal Vector Magnitude, Energy, Entropy, FFTPeak, and Standard Deviation were then selected.	Bayes,SMO, IBk, KStar, Multiclass Classifier, Bagging, DT, J48 and RF	RF 85%
[79]	IMU/ Lower back, both sides of the thigh, shank, foot	10m gait	11SP, 9NDP [4]	Data were filtered with a fourth-order bi-directional Butterworth band-pass filter, then minimal peak distance and minimal peak height were applied to the resulting data. after that different gaits parameters were computed.	RF, Adaboost, DT, Gaussian naive bayes, MLP	The shank placement DT 89.13%
[80]	IMU/ shank	10 m gait	8SP,7AB	Data were normalised and labeled different gait phases	MLP	99.35%

[4] NDP: Neurologically disordered participants

[81]	IMU, EMG,temperature/ Arms and chest	Flexor synergy, shoulder flexion hand to lumbar pronation and supination	NA	Employed Empirical mode decomposition [184] to partition the times series data into the three first intrinsic mode functions. The mean values and standard deviations of these components are used in conjunction with mean values and standard deviations, entropy and energy of the motion signals as features for large joint actions.	AdaBoost	Accuracy of 99.25%.
[82]	IMU,/ Arms, forearms, thighs	ADLs (e.g. walking, walking up/downstairs, arm and leg flexion/extension, arm rotation, writing, using phone, drinking)	NA	walking-related gait parameters (stride duration, cadence and stride count)	unspecified regression models	
[83]	IMU,/ Arm, forearms, hand	Flexion/extension of the elbow, supination/pronation of the forearm, extension/flexion of the wrist	13AB, 13SP	Linear interpolation was done to synchronise data, then data was normalised	Different KNN models, Different SVM models, Fine tree	Fine KNN 98.5%
				Clinical assessment emulation		
[84]	Accelerometer/ Wrist and the sternum	tasks associated with WFT	34 AB	Segmented time-series data	RF	correlation with therapists scores R^2=0.97
[85]	IMU/ Forearm	ADLs (Doing the laundry, Performing kitchen activities, Shopping, Making the bed.)	10 AB, 10 SP	Extracted entropy, mean, and variance-based measures	Tree based	Accuracy of 88%
[86]	IMU/ Arms and chest	A battery of activities from WMFT	16 SP	Derived the time-series magnitudes of displacement, velocity, acceleration, and jerk to extract multiple time series features i.e.: minimum, maximum, and mean values, root mean-square value, ratio of the magnitude of the dominant frequency and total signal energy, jerk, skewness, signal entropy, kurtosis, correlation coefficients computed for different axes, and duration of the data segments.	RF	0.38 RMSE

[87]	Accelerometer, flex/ Shoulder, elbow, wrist, finger	Seven different exercises based on the short FMA	24SP	Raw sensor data was denoised with 5 point smooth method, and AMP, MEAN, RMS, JERK, and ApEn were extracted and then RRelief algorithm was applied to find the optimal features for each exercise.	ELM,SVM	SVM 92.2%
[88]	IMU/ Sternum, arms, wrist, elbow	Synergy, out of synergy, combination of synergies, wrist/hand function and fine motor coordination	8SP	RMS, mean, entropy, dominant frequency.	DT, Bagging Forest	Bagging Forest reported lowest RMSE
[89]	Accelerometer/ arms, shanks	ADLs (Exercises from the Oxford Grading Motor Scale)	4SP	Gravity component was removed from the norm of the acceleration data then the mean, max, mean, normalized average rectified jerk, powers and frequencies of FFT	SVM	82%
[90]	Accelerometer/ Wrist	ADLs (Exercises from the Oxford Grading Motor Scale)	59SP	signal vector magnitude is computed by substracting the gravity effect from the acceleration, then DWT to extract wavelet coefficients, normalised Sum of Absolute value of DWT coefficients is used as features	LMGP, lSVM, rbf SVM, mlp	LMGP reached RMSE 3.12 for Chronic) and 5.75 for acute
[91]	IMU/ Wrist	grabbing a cube and moving it for an ARAT assesment	34SP	Raw data from IMUs	Matching pursuit	Accuracy of 95 percent
[92]	IMU/ Wrist,sternum	continuous, random, voluntary upper-limb movements spanning the entire range of active motion	23SP	Zero-Crossing Decomposition applied on gravity free acceleration, resulting data is normalised to engineer different features	unspecified regression model	R^2 value of 0.985
[93]	IMU/ Wrist, and feet	Stretch and hold their arms for 20 seconds, and lift and stretch their left or right leg	15SP	Features related to the degree of drift of the limbs	Ensemble algorithm and SVM	Accuracy of 83.3% for SVM
[94]	IMU/ hands, arms, Knee, Tibia	ADL	120SP	Features related to time domain	Logistic regression	Overall accuracy of 79.3% for SVM

[95]	IMU and flex sensors/ Wrist	A variety of limb movements, including joint movements, collaborative actions, stability exercises, and coordination tasks, are encompassed in the provided text. These movements span both upper and lower limbs and involve different quantities of actions for each category.	21SP	Features related to time domain	SVM	95%

Table A.2: Summary of the reviewed papers

[60]	Real-time gesture recognition performance to control a five finger dexterous robot;	• Focused only on user-specific condition, where the training data and the verification data are from the same subject posing a generalisation issue. • Did not test the model on stroke patients. • There is no mention if the final prototype has been used in clinical environments afterwards.
[61]	Monitor the overall body activity and the drinking activity from the liquid level of the mug. Subjects were asked to accomplish while holding the cup some ADL. the resulting data were fused together to increase the performance of the processing algorithm.	• The absence of a study on the acceptability of the smart cup by stroke patients. • No stroke patients were included and no research studies were conducted. • The absence of an assessment system.
[62]	A single sensor was used to collect data from the impaired arm of stroke survivors, the participants execute twenty different arm tasks in two different environment settings: patients at home and patients at labs.	• The absence of a real-time implementation of the system • There is no mention if the final prototype has been used in clinical environments afterwards.
[63]	The study determined signal features that are best suited for activity recognition with various populations (stroke patients, able bodied and elderly participants) independent of the chosen classifier.	• The study did not present any platform. • The classifiers were not customized to the specific HAR application.
[64]	Developed a home-based rehabilitation system that can recognise and record the type and frequency of rehabilitation exercises conducted by the user using a smartwatch.	• The total number of patients who completed the program was relatively small to derive statistically strong evidence. • The actual accuracy of exercise detection at home was not assessed.
[65]	A system that classifies functional and nonfunctional arm movement from accelerometry sensor data.	• Limited activities and ADL tasks that the participants performed..
[66]	Compared HAR performance for persons with stroke while varying the origin of training data, based on either population (AB or SP) or environment setting.	• The healthy cohort did not age match the stroke cohort. • Different sensor placements throughout the study. • Data associated with stroke patients in home setting was small compared to the others.

Paper	Study aim	Limitation
[67]	Proposed a wearable activity monitoring system based on a fuzzy logic based activity classifier that exploits fused information from the sensors which accounts for behavioral constraints and estimates the body elevation during standing and locomotion.	• Non-uniformity of the number of data samples for the different activities. • Limited number of SP. • There is no mention if the final prototype has been used in clinical environments afterwards.
[68]	Developed a novel prediction model based on ML algorithms and determine the accuracy of detecting different ADLs performed by stroke survivors. The study was conducted in a simulation living room and kitchen. Lastly, additional independent training and testing data were collected to perform external validation to further imitate real-world prediction conditions.	• The sample size is relatively small so the model might not generalise well. • Data was collected in a semi-naturalistic environment instead of participants' homes. • The accuracy might be more improved. • It is a pilot study so it did not yield to an assessment platform yet.
[69]	Encoded time-series data into gray-scale and RGB images and tested different CNN models to profit from computer vision development.	• No SP were included in the collected data. • The presence of confounding movements induced by clinical practitioner patient interactions while performing the exercises.
[70]	A comparative study to investigate the performance of different sensors and different placements for classifying four different ADLs with the purpose to find the optimal placement of a single sensor that achieves best accuracy.	• The study was preliminary. • Only five ADLs were included, they also have similar patterns • There is no mention if the final prototype has been used in clinical environments afterwards.
[71]	A new dataset was provided for real-time activity recognition of stroke-affected patients, encompassing six frequent recognition activities and 20 common features were extracted. Activities were recognized in real-time using a standard LSTM-RNN, and the recognised activities were presented as text and sound output through a MATLAB application.	• Accuracy can be improved. • Accelerometer did not model well the movements.
[72]	Used two sensors to differentiate between goal-directed exercises and ADL as well as detecting the poorly executed exercises following the FMA [97] assessment during in-home rehabilitation exercises.	• The sample size was relatively small and thus the reported results may not be generalised. • Movements that were both goal-directed and non goal-directed in nature were not considered. • There is no mention if the final prototype has been used in clinical environments afterwards.

Paper	Study aim	Limitation
[73]	The Establishment of a quantitative measurement system of the amount of hand use of 11 motor tasks of ADL using two sensors.	• The proposed technology cannot capture the use of the hands for stabilizing objects (e.g., holding a cup or stabilizing a piece of steak with a fork) as it focuses on estimating the amount of hand movement. • No stroke patients were included and no research studies were conducted. • There is no mention if the final prototype has been used in clinical environments afterwards.
[74]	Developed an approach that identifies and counts functional primitives that constitute rehabilitation activities in an automated manner.	• The primitives were only recognised, no system has been set up to count them. • No platform was implemented. • Has not been tested in clinical settings.
[75]	SmartPants is used to perform therapy exercises and recognise some ADL lower-limbs taks.	• The number of sensors might be reduced to design a more unobtrusive system. • Did not test the model on stroke patients. • There is no mention if the final prototype has been used in-clinical environments afterwards.
[76]	Validate a general probabilistic modeling approach for the classification of different pathological gaits.	• Metrics for gait assessment were not included.
[77]	Recognise purposeful and non purposeful arms' movements of post-stroke patients when performing ADLs for identifying and promoting the use of the impaired limb during daily life in people affected by stroke. different datasets were investigated to see which gives better results, namely SP, AB, and both.	• Data collected from index and wrist sensors only. • the recruited groups were not age-matched. • Data were collected in a controlled environment.
[78]	Investigated the performance of unimanual, bimanual asymmetric, and bimanual symmetric tasks in participants post-stroke and controls for a variety of signal processing and ML tools. The system classifies bi-manual and uni-manual tasks.	• Accuracy could be further improved. • There is no mention if the final prototype has been used in clinical environments afterwards. • Accuracy could be further improved. The sample size was relatively small and thus the reported • Results may not be generalised.
[79]	Examined IMU sensor placement configuration with different classification algorithms and differentiate between SP and NDP gaits. It showed that shank placement provided better accuracy.	• Limited sample size. • No clinical application was implemented from this research yet.

Paper	Study aim	Limitation
[80]	Developed a model that can recognise stroke gait with the help of therapists.	• Limited number of participants which causes the system to not generalise well. • There is no mention if the final prototype has been used in clinical environments afterwards.
[81]	A sensing sub-system placed on a shirt sleeve (smart shirt) collected data that are then processed locally on a smart wireless access point based on Raspberry Pi and then sent to an Android device via a Transmission Control Protocol (TCP) socket by a Wi-Fi master node where the patient is given a personal account that the physicians use to login into and visualise the real-time data. The information is then sent to a data cloud built with MySQL where it is stored and then pushed to a computing cloud that utilises ML algorithms implemented on MATLAB to evaluate the data [97]..	• Limited number of activities included. • The data was collected from a single person. • No stroke patients were included and no research studies were conducted. • There is no mention if the final prototype has been used in clinical environments afterwards.
[82]	Proposed three digital biomarkers namely convergence points, physical activity and functional range of motion for the longitudinal performance monitoring and movement evaluation of stroke patients	• Some subtle movement changes require further research to distinguish improved movement ability due to recovery from movement compensation mechanisms. • Further analysis should be done to see how the algorithm would generalise.
[83]	Evaluate the feasibility of using body-worn sensors to track rehabilitation exercises in the inpatient setting and counting exercise repetitions in order to identify stroke severity	• Has only been tested on three basic ADLs. • gyroscope data did not include all patients. • Stroke patient data were from subjects with at least moderate strength and did not include more severe cases.
[84]	Only two IMU sensors to assess quality of movement Functional Ability Scale scores [99], the results were then correlated with therapists scores giving very high accuracy.	• Further analysis should be done to see how the algorithm would generalise. • No stroke patients were included and no research studies were conducted. • There is no mention if the final prototype has been used in clinical environments afterwards.
[85]	A single sensor to measure upper extremities functional use during ADL and distinguish it from the upper extremities movements that occur while walking.	• Establishing clinical validity requires further research with larger patient populations to determine how well this methodology generalises across stroke survivors.

Paper	Study aim	Limitation
[86]	Multiple upper-limbs assessment system utilising two different rehabilitation evaluation scoring systems the FAS [185] and FMA associated with different ADL tasks.	• Further analysis on supplementary participants to see how the algorithms would generalise. • Accuracy could be improved further. • Has not been tested in clinical settings.
[87]	Proposes a novel remote quantitative FMA assessment system in home settings. Sensors record the movement information in real time and wirelessly transmit to the computer through ZigBee protocol and finally upload to the web server database through Internet.	• Only seven exercises were included. • The placement of sensors has not been investigated further. • The system is obtrusive. • There is no mention if the final prototype has been used in clinical environments afterwards.
[88]	Evaluated two approaches designed to estimate the quality of post–stroke upper extremity motion as measured by the FMA subscale for the upper extremity using paretic and non–paretic limb kinematic data.	• Limited number of participants which causes the system to not generalise well. • No clinical application was implemented from this research. • Research was not conducted in a research environment.
[89]	Assessed whether long-term monitoring of seven days or more, in unilaterally impaired stroke patients is useful in determining motor impairment using [102].	• Very small sample size which would not generalise to more data. • The presence of confounding movements induced by clinical practitioner patient interactions while performing the exercises.
[90]	Developed an automated system that can predict the assessment score in an objective manner to do so two new features were proposed.	• There is no mention if the final prototype has been used in clinical environments afterwards. • Included activities are limited
[91]	Employ time-frequency methods to provide a better analytical basis for the derivations.	• Very small data sample, only 78 data segments were collected. • There is no mention if the final prototype has been used in clinical environments afterwards. • No clinical application was implemented from this research. • Research was not conducted in a research environment.

Paper	Study aim	Limitation
[92]	Proposed an approach to estimate upper-limb impairment in stroke survivors using two wearable inertial sensors, on the wrist and the sternum.	• Small sample size which may not generalise to more data. • Reliance on the performance of large, continuous movements, which can be tiresome for stroke patients.
[93]	Developed an autonomous grading system for stroke patients using NIHSS and MRC scores.	• Very small sample size which may not generalise to more data. • Very limited set of exercises.
[94]	A wearable motion capture system, employing nine-axis IMUs and Flex sensors, recorded real-time rehabilitation data from stroke patients. This data was used to establish a rehabilitation assessment model akin to the clinical FMA score, offering quantitative rehabilitation scores through sensor fusion and machine learning.	• The sensitivity of signal processing to the initial placement of wearable sensors, potentially leading to deviations in rehabilitation assessment outcomes. • The influence of body compensation on assessment accuracy in more severe patients was noted. • Further improvements maybe sought, using more advanced algorithms (e.g: deep learning).
[95]	Data on stroke patients were collected using two inertial sensors attached to their wrists, with ARAT task and total scores estimated through supervised machine learning. It is hypothesized that this approach can yield ARAT score estimates with an error similar to or smaller than clinically relevant changes, enabling automated administration independent of expert input, thanks to the straightforward setup of just two wearable sensors.	• A small sample size was employed, potentially affecting the prediction accuracy and model robustness. Expanding the sample size with greater diversity may enhance the model's performance. • The reliance on clinical scores as reference data for machine learning training introduces inherent limitations, as the model merely reproduces information from these scores, which may not capture movement quality comprehensively.