Deep Learning for Gait Prediction: An Application to Exoskeletons for Children with Neurological Disorders

Rania Kolaghassi

A Thesis Submitted to the University of Kent for the Degree of Doctor of Philosophy in Biomedical Engineering

School of Engineering
University of Kent

August 2023

To my parents, who dedicated their lives to us, I dedicate this to you.

Abstract

Cerebral Palsy, a non-progressive neurological disorder, is a lifelong condition. While it has no cure, clinical intervention aims to minimise the impact of the disability on individuals' lives. Wearable robotic devices, like exoskeletons, have been rapidly advancing and proving to be effective in rehabilitating individuals with gait pathologies.

The utilization of artificial intelligence (AI) algorithms in controlling exoskeletons, particularly at the supervisory level, has emerged as a valuable approach. These algorithms rely on input from onboard sensors to predict gait phase, user intention, or joint kinematics. Using AI to improve the control of robotic devices not only enhances human-robot interaction but also has the potential to improve user comfort and functional outcomes of rehabilitation, and reduce accidents and injuries.

In this research study, a comprehensive systematic literature review is conducted, exploring the various applications of AI in lower-limb robotic control. This review focuses on methodological parameters such as sensor usage, training demographics, sample size, and types of models while identifying gaps in the existing literature.

Building on the findings of the review, subsequent research leveraged the power of deep learning to predict gait trajectories for the application of rehabilitative exoskeleton control. This study addresses a gap in the existing literature by focusing on predicting pathological gait trajectories, which exhibit higher inter- and intra-subject variability compared to the gait of healthy individuals. The research focused on the gait of children with neurological disorders, particularly Cerebral Palsy, as they stand to benefit greatly from rehabilitative exoskeletons. State-of-the-art deep learning algorithms, including transformers, fully connected neural networks, convolutional neural networks, and long short-term memory networks, were implemented for gait trajectory prediction. This research presents findings on the performance of these models for short-term and long-term recursive predictions, the impact of varying input and output window sizes on prediction errors, the effect of adding variable levels of Gaussian noise, and the robustness of the models in predicting gait at speeds within and outside the speed range of the training set.

Moreover, the research outlines a methodology for optimising the stability of long-term forecasts and provides a comparative analysis of gait trajectory forecasting for typically

developing children and children with Cerebral Palsy. A novel approach to generating adaptive trajectories for children with Cerebral Palsy, which can serve as reference trajectories for position-controlled exoskeletons, is also presented.

Acknowledgments

"My Lord, benefit me with what you have taught me, teach me that which will benefit me, and increase my knowledge."

Prophet Muhammad (PBUH)

I'm grateful to God for the privilege of being on this journey, for guiding me through it, and for all His blessings.

I extend my heartfelt appreciation to my supervisors Prof. Konstantinos Sirlantzis and Dr Gianluca Marcelli, who encouraged me to continue my studies. Thank you for entrusting me with this research, and for providing me with your valuable knowledge, guidance, and insights. I've learned a lot from you and I deeply appreciate your support and motivation throughout the years.

I'm also grateful to my external examiner, Prof. Aladdin Ayesh, my internal examiner, Dr Moinul Hossain, as well as Dr Viktorija Makarovaite and Dr Sanaul Hoque for their feedback on my work.

I would like to thank my collaborators from the M.O.T.I.O.N. project, whom I'm very fortunate to be part of. I would like to particularly thank Dr Mohamad Kenan Al-Hares, Mohammadhadi Sarajchi, Markus Hunt, and Dr Sarah Crombie.

I'm thankful to everyone who supported me in my journey, at Kent and beyond, including all the researchers whom I read their books and articles; I might not know you personally, but your work definitely enriched mine.

I would also like to thank my family, especially my grandmother Amnah, my grandfather Ghassan, who sadly passed away shortly before seeing this thesis come to completion (may God have mercy on him), and my brother Rashid. To my Uncle Luay, I still remember your joyous reaction when I told you about my desire to undertake this research. Your unconditional love and support, in academics and beyond, mean the world to me.

To my friends and colleagues, who never failed to brighten my days, I will forever cherish my memories with you. Thank you for all the lunches, dinners, and coffee breaks in between, for all the laughs we had, the gym sessions we tried to maintain,

and for being there for me. Special thanks to Dr Majd Alluhaymaq, Chantal Zorzi, Pavlos Nicolaou, Paula Delgado-Santos, Dr Luma Tabbaa, Dr Rawan Allouzi, Badea'a Al-Sukhni, Jenna Seetohul, Dr Rukyie Kaya, Dr Marco Santopietro, Hiba Jawharieh, Khawla Alhasan, Dr Ayda Majd Ardekani, Hazal Su Bıçakcı, and to many others whom I'm very fortunate to have crossed paths with throughout this degree. I'm truly grateful to have met you all.

Above all, I would like to thank my parents, Nasreen Al-Shawwa and Bassam Kolaghassi. I will never be able to thank you enough for everything you have offered me throughout my life. I'm indebted to you, for your unconditional love, patience, sacrifices, motivation, guidance, prayers, and for so much more. Everything I have accomplished or will ever achieve is because of you.

Publications

The following publications have resulted from this research work:

Journal Articles:

1. **R. Kolaghassi**, M. K. Al-Hares, and K. Sirlantzis, "Systematic review of intelligent algorithms in gait analysis and prediction for lower limb robotic systems," IEEE Access, vol. 9, pp. 113 788–113 812, 2021.
2. **R. Kolaghassi**, M. K. Al-Hares, G. Marcelli, and K. Sirlantzis, "Performance of deep learning models in forecasting gait trajectories of children with neurological disorders," Sensors, vol. 22, no. 8, p. 2969, 2022.
3. **R. Kolaghassi**, G. Marcelli, and K. Sirlantzis, "Deep learning models for stable gait prediction applied to exoskeleton reference trajectories for children with cerebral palsy," IEEE Access, vol. 11, pp. 31 962–31 976, 2023.
4. **R. Kolaghassi**, G. Marcelli, and K. Sirlantzis, "Effect of gait speed on trajectory prediction using deep learning models for exoskeleton applications," Sensors, vol. 23, no. 12, 2023.

Conferences:

1. **R. Kolaghassi**, M. K. Al-Hares, G. Marcelli, and K. Sirlantzis, "Prediction of Gait Trajectories of Children with Neurological Disorders using Intelligent Algorithms for Exoskeleton Control", BioMedEng 2022, UCL. (Poster)

Table of contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
BPNN	Back propagation neural network
CLME	Complementary limb motion estimation
CNN	Convolutional neural network
CP	Cerebral palsy
DL	Deep learning
DT	Decision trees
DTW	Dynamic time warping
EEG	Electroencephalogram
EMG	Electromyogram
FCN	Fully connected neural network
GMFCS	Gross Motor Function Classification Scale
FS	F-score
FSM	Finite state machine
FSR	Foot pressure sensors
GRF	Ground reaction force
HMM	Hidden Markov model
IMU	Inertial measurement unit
LDA	Linear discriminant analysis
LRP	Layer-Wise Relevance Propagation
LSTM	Long-short term memory
MAE	Mean absolute error
ML	Machine learning
MLP	Multilayer perceptron
MSE	Mean squared error
PAFO	Powered ankle foot orthosis
PCA	Principal component analysis
RF	Random forest
RL	Reinforcement Learning
RMSE	Root mean squared error
RNN	Recurrent neural network
RQ	Research question
SVM	Support vector machine
TBE	Time based estimation
TD	Typically developing
XAI	Explainable artificial intelligence

Chapter 1: Introduction

1.1 Background

Cerebral Palsy (CP) is a neurological condition characterised by motor impairments resulting from cerebral injury that occurs shortly before, during, or after birth [1]. CP is the most common childhood motor disability, affecting between 1 to 4 children per 1000 live births worldwide [2]. Although CP is non-progressive, it is a lifelong condition. Since it cannot be cured, the goal of clinical intervention is to minimise the impact of disability on the individual's life [3].

CP manifests itself in various forms. In addition to motor impairments, CP can lead to intellectual impairments (in about two-thirds of patients), neurological abnormalities that affect hearing and vision, seizures, and gastrointestinal problems [4]. Motor impairments associated with CP are often classified into three main types: spastic, dyskinetic, and ataxic CP. Spastic CP which accounts for 70% to 80% of the cases, is characterized by deep tendon reflexes, hypertonicity (increased muscle tone), tremors, and a crouch gait [4]. Dyskinetic CP, comprising 10% to 20% of the cases, is characterised by abnormal movements and posture resulting from impairments in movement control, muscle tone, and coordination [4, 5]. Ataxic CP, the least common type at 5% to 10%, is characterised by impairments in balance and coordination, and a wide-based gait [4].

Various clinical interventions are available for managing CP, including medications such as botulinum toxin injections that assist in muscle relaxation, surgical interventions like

selective dorsal rhizotomy to reduce spasticity, and external aids such as orthoses that restrict undesirable joint movements [4, 6]. Physiotherapy and rehabilitation are also key interventions. In the case of children with CP, physiotherapy and rehabilitation play an important role in increasing muscle strength and local muscular endurance, as well as maintaining or increasing joint range of motion [7, 8]. Improvement in gait patterns is achievable with those interventions and positively correlates with increased independence and community participation [9, 10].

There has been a notable increase in the development of novel robotic devices aimed at the rehabilitation of pathological gait, including for children with CP [11]. Among those devices, exoskeletons have gained attention as a type of robot-assisted therapy. Exoskeletons are comprised of a ridged structure with sensors, actuators, and a control strategy that provides assistive torque to joints, to help with walking [12]. Exoskeletons have aided in 'massed practice', a training approach in which patients perform exercises with reduced number and duration of breaks in-between, in one rehabilitation session [13, 14]. Massed practice can optimise motor learning while reducing the strain on therapists [13].

Several studies have demonstrated that exoskeletons contribute to favourable rehabilitation outcomes, including improved spatiotemporal parameters (increased mean velocity, cadence, and stride length), decreased energy expenditure, and improved hip and knee extension during the stance phase of the gait cycle [13, 15]. However, the extent of benefits has varied across these studies, potentially due to differences in methodological setup and practice time [13, 15]. Additionally, the effectiveness of the exoskeleton may depend on the control strategy, which can vary widely.

The aim of a rehabilitative exoskeleton can be to provide assistive control, challenge-based control, or adaptive control [16]. However, there is currently no consensus on the optimal control strategy, as it may depend on the individual's condition and their needs [15, 16]. Nonetheless, improving the control strategy of an exoskeleton, as well as enhancing human-robot synchronization, are essential for optimising rehabilitation outcomes. Exoskeleton control strategies are typically a three-level hierarchy, starting from the high-level which detects user intention and desired locomotion, to the mid-

level which translates desired movement into the necessary actions, and finally, the low-level which implements the selected action [17]. Further research and development efforts can be focused on improving all three stages.

Accurate and timely estimation of user intention, which is part of the high-level control of an exoskeleton can result in smooth human-robot interaction. In the field of robotics, *intention* refers to 'the need for the robot to have knowledge of some aspect of the human's planned action in order for the robot to appropriately assist toward achieving that action' [18]. Intention can be decoded into discrete states, which trigger the start of certain movements or transitions between discrete control modes, as well as continuous states such as the desired position trajectory [18]. Recent advances in exoskeleton control have leveraged the power of artificial intelligence (AI) for intention prediction and for the high-level control of exoskeletons [19].

1.2 Research Motivation

Artificial intelligence and deep learning methods have experienced growing utilization in gait analysis [20,21]. This includes diagnosing movement disorders such as Parkinson's disease [22] or Cerebral palsy [23], identifying humans for biometric applications [24], preventing sports injuries [21], and for sports management [21].

There are several advantages to using artificial intelligence in the context of exoskeleton control, particularly deep learning (DL) methods.

- Outperforming conventional methods: DL outperformed heuristic-based approaches and rigid mathematical models defined by experts for tasks such as gait event classification, achieving higher accuracy and shorter duration between the time of event prediction and the time of the true gait event [25]. Timely predictions are necessary for providing timely support during gait events.
- Human motion intention prediction: DL enabled the prediction of human motion intention, resembling the user's desired motion for the exoskeleton to follow, and utilising bio-signals such as EEG and EMG [26–28]. This has led to a reduction

in the mean absolute error as well as the average delay between the movement of the human and the robot [29].

- Accelerated exoskeleton control development: DL can expedite exoskeleton development by allowing the use of raw spatiotemporal gait data as input, eliminating the need for extensive data pre-processing or feature engineering that are time-consuming and require expert selection. Extensive data pre-processing and feature engineering may sometimes lead to discarding features that may be otherwise important for describing human movement and analysing gait [30].

- Data-driven predictions: DL methods also generate data-driven predictions, utilising large amounts of data captured from low-cost and non-intrusive wearable devices including smartwatches, inertial measurement units (IMUs), and smartphones [21]. They can deal with high-dimensional data and interdependent gait parameters [31, 32].

- Data-fusion (multi-joint and multi-modality approaches): DL methods allow for data fusion, including multi-joint data fusion where predictions are generated based on data from several joints (e.g., hip angles and knee angles) and multi-modality data fusion where predictions are generated based on input from multiple sensors (e.g., plantar pressure combined with acceleration data). Multi-modality models demonstrated superior performance over single-modality models [33].

- Personalised/individualised treatment for each patient: DL has the potential to generate personalised/individualised treatments for each patient, particularly advantageous in rehabilitative exoskeletons. Studies have reported DL network abilities in capturing the variability of humans [32], while also being capable of individualising to specific individuals [34]. The ability of deep learning models to customise the reference trajectory an exoskeleton follows, based on user parameters such as anthropometrics and gait speed, is key for enhancing synchronised and seamless human-robot interaction and may lead to improved functional outcomes of rehabilitation [35, 36].

This research was driven by the growing importance and value of developing rehabilita-

tive exoskeletons for children with Cerebral Palsy. Considering the promising potential demonstrated in the literature regarding the integration of deep learning for exoskeleton control, this study focuses on implementing deep learning models for the supervisory control of exoskeletons designed for children with Cerebral Palsy.

1.3 Research Aims and Objectives

The aim of this study is to develop and advance AI systems that improve the control of rehabilitative exoskeletons, specifically those targeting children with neurological disorders, with a focus on Cerebral Palsy. This target group stands to benefit greatly from exoskeleton technology, and the study aims to explore the use of artificial intelligence (AI) for this purpose.

The objectives of this research work are as follows:

- Identify various types and applications of AI models used in the literature for lower limb robotic control.
- Explore the use of deep learning methods for predicting trajectories of children with neurological disorders. These predictions can serve as feed-forward input for exoskeleton controllers to compensate for control time delays or act as reference patterns for position-controlled exoskeletons.
- Investigate different characteristics of state-of-the-art deep learning models and how their performance is impacted by variables such as input and output length, the addition of noise, and robustness in predicting gait at varying speed ranges, both within and outside the range of gait speeds used for training.
- Explore approaches to generating adaptive and individualised gait patterns for children with Cerebral Palsy.

Specifically, this research work aims to address the research questions listed below.

1.4 Research Questions

This thesis attempts to address the following research questions (RQ):

- RQ 1: Can deep learning models accurately predict the gait patterns of individuals with pathological gait such as children with Cerebral Palsy?
- RQ 2: What is the effect of longer time horizons for gait trajectory predictions regarding robustness and accuracy especially considering the effects of noise?
- RQ 3: What are possible methods for improving deep learning models regarding the propagated error in long-term predictions?
- RQ 4: Can deep learning models trained on gait data from typically developing children be utilized to generate continuous and adaptive target/reference trajectories for children with Cerebral Palsy?
- RQ 5: What is the effect of different gait speeds in training deep learning models for gait trajectory predictions?

1.5 Key Contributions

The main contribution of this research is the development and advancement of AI systems primarily designed and optimised for controlling rehabilitative exoskeletons targeting children with neurological disorders, particularly Cerebral Palsy. The outcomes of this research study can be summarised in the following points, each linked to the research questions (RQ) they address:

- Reviewed the various applications of AI and its integration in the supervisory level of control for lower limb exoskeletons. This review identified gaps in the existing literature and provided insights into future research directions.
- Implemented and compared the performance of several state-of-the-art deep learning models for gait trajectory prediction.

- Investigated the feasibility of forecasting joint kinematics of children with neurological disorders, which exhibit high inter- and intra-subject variability compared to healthy gait patterns, using deep learning. (RQ 1)
- Extended our understanding of the influence of changing the length of input and output window sizes on gait trajectory prediction and the influence of the addition of variable levels of Gaussian noise on the stability of predictive models. (RQ 2)
- Introduced an approach for optimising the stability of long-term forecasts, addressing challenges associated with prediction accuracy over extended time periods. (RQ 3)
- Proposed a methodology for generating adaptive gait trajectories for children with Cerebral Palsy, which can serve as reference trajectories for position-controlled exoskeletons. (RQ 4)
- Explored the robustness of the models in predicting joint kinematics at varying gait speeds that are within and outside the training set speed range. (RQ 5)

Several publications have been produced as a result of this research work. A full list of those publications can be found on page v. Chapters of this thesis have been based on those papers. The author of this thesis is the first author of those papers.

1.6 Thesis Structure

The structure of the thesis is as follows:

- *Chapter 2* presents the findings of a systematic literature review on the use of AI for controlling lower limb robotics, including exoskeletons and orthoses. In this chapter, key information regarding machine learning and gait analysis was introduced. The review aims to explore the application of AI in predicting or detecting key parameters related to gait, such as gait phase, locomotion mode, joint kinetics, and joint kinematics. Throughout the review, the focus was on examining several methodological parameters including sensor usage, training demographics, sample size, and types of models employed in the studies. Additionally, we

identified gaps in the existing literature and highlighted areas for future research and development.

- *Chapter 3* introduces the three datasets used to develop (train and evaluate) the trajectory forecasting models presented in Chapters (4 to 6). It also discusses ethical considerations regarding the data used.

- *Chapter 4* presents the results of a study that explored the feasibility of forecasting the trajectories of children with pathological gait using deep learning models. In this chapter, the performance of deep learning models, namely a long-short-term memory network and a convolutional neural network, was explored and compared against non-intelligent methods. The impact of varying the length of input and output window size on the accuracy of predictions was also investigated.

- *Chapter 5* focuses on the prediction of gait trajectories for two distinct groups: typically developing children and children with Cerebral Palsy. This chapter examines the performance of the deep learning models in both short-term and long-term prediction of joint kinematics. Additionally, an approach to optimise the stability of long-term predictions was introduced. The impact of introducing variable levels of Gaussian noise on the stability of the models was investigated as well. The chapter proposes a novel approach for generating continuous adaptive trajectories for children with Cerebral palsy, that serve as reference patterns for position-controlled exoskeleton.

- *Chapter 6* investigates the performance of deep learning models in predicting joint kinematics across a range of gait speeds. The study specifically focuses on assessing the robustness of these models in predicting gait patterns at speeds both within and outside the training gait speed range.

- *Chapter 7* summarises the overall thesis, emphasizing key contributions and insights derived from the conducted studies. Limitations of this research work were discussed and potential future research directions were identified.

Chapter 2: Literature Review on Intelligent Algorithms for Exoskeleton Control

2.1 Overview

In this chapter, a systematic literature review on the use of AI for the control of lower limb robotics, specifically exoskeletons and orthoses, is presented. The review aims to explore the diverse applications of AI in predicting various gait parameters. Methodologically, this review examines the types of models employed, sensor usage, and training demographics in the existing literature. Limitations in current implementations are discussed and future research directions are also proposed.

2.2 Introduction

The development of wearable robotics devices, such as exoskeletons and orthoses, has gained significant attention in recent years. These devices serve several functional and medical purposes. An exoskeleton, in particular, is an electro-mechanical device comprised of actuators, sensors, and controllers that provides torque to joints [12]. The provision of supportive torque allows for physical actions to be performed with more ease and lower strain. The concept of an exoskeleton dates back to the 1960s when they were initially designed and developed for military use [12]. Hardiman I exoskeleton, developed by the General Electric Company, aimed to augment the endurance and strength of the soldiers, leading to what was described as the 'union of man and

machine' [37]. Over the years, exoskeletons evolved to serve a broader range of purposes including industrial applications [38], rehabilitation, and restoration of gait for patients with Spinal Cord Injuries [39], Cerebral Palsy [40], and Multiple Sclerosis [41]. These exoskeletons are designed to support the upper limbs, lower limbs or the entire body and can be categorised as passive, active or quasi-passive [42]. Several state-of-the-art exoskeletons exist in various domains. For industrial applications, Berkeley Lower Extremity Exoskeleton (BLEEX) is one example. It is a seven-degree-of-freedom exoskeleton, actuated with linear hydraulics enabling heavy load lifting [43]. The MIT exoskeleton is another quasi-passive exoskeleton designed for heavy load lifting. It utilises springs and dampers instead of actuators and acts as an intermediator, transferring 80% of the load from the person to the ground [44]. In the field of rehabilitation, Hybrid Assistive Limb (HAL-5) developed by the University of Tsukuba, is a full-body exoskeleton that targets healthy people as well as patients. It enables paraplegics to walk by decoding their intentions [45]. Another example is MINDWALKER, which uses brain-generated electroencephalogram (EEG) and electromyogram (EMG) signals to control a series of elastic actuated full-body exoskeleton targeted for paraplegics [46].

Orthoses are another category of assistive and corrective technologies, sometimes used interchangeably with exoskeletons. However, there is a difference between the two. According to Herr, the purpose of orthosis is to assist those with pre-existing pathologies, contrary to exoskeletons which augment human capabilities, including healthy people [47]. Orthoses can also be passive [48] or active [49]. In the MIT labs, an active ankle-foot orthosis for drop-foot gait treatment has been developed [49].

Lower limb exoskeletons, when used as assistive devices, have two primary applications: (i) rehabilitation and gait training, or (ii) locomotion assistance to help perform daily life activities [50]. The control strategy, which specifies the way the exoskeleton moves and interacts with the user, is therefore based on the exoskeleton's application, as well as the condition of the patient using the device. Trajectory tracking is a type of control strategy, whereby an exoskeleton allows the patient to walk following a pre-defined gait trajectory pattern, often obtained from a healthy person. Assist as needed, is another control strategy whereby the support given by the exoskeleton is variable and dependent on the user's need. The level of assistance provided may be dependent on

phase of gait, the level of effort exerted by the patient, and the stage a patient is in their recovery journey [50].

The control scheme is often a multi-level hierarchy, consisting of low, mid and high levels of control. The high-level control is responsible for user intention detection/event estimation. The mid-level control is responsible for exoskeleton state transitions, based on the intention/event detected. An example of a mid-level controller is a Finite State Machine (FSM). The low-level control is responsible for tracking user motion and ensuring stability, often utilising force, position, or impedance control [51–53]. Intention detection and gait event estimation have a crucial role in the exoskeleton's control and functionality as they are at the top of the control hierarchy. Various techniques have been used to analyse gait for their estimation. Controlling exoskeletons is one of the most recent uses of gait analysis, which has already been used in various clinical and non-clinical settings. In clinical settings, gait analysis is used for rehabilitation assessment [54], and diagnosis of pathology [55]. In non-clinical settings, gait analysis is used in sports for post-injury recovery monitoring [56] and performance evaluation [57], in security, for biometric identification and authentication [58–60], in safety, for elderly fall detection technologies [61], and in wellbeing, for emotional state identification [62].

Gait analysis involves measuring or estimating a range of parameters, including spatial-temporal parameters, EMG activity, kinematic, and kinetic parameters during walking or other locomotion activities [63].To perform this analysis a range of wearable and non-wearable sensing modalities are used. Wearable sensors include inertial measurement units (IMUs) with accelerometers and gyroscopes, goniometers, electromyography (EMG), electroencephalographs (EEG), and foot pressure sensors. Non-wearable sensors include ground reaction force (GRF) plates and motion capture systems [63], [20].

The measurements obtained from the wearable sensors are then processed to derive gait parameters. Researchers have experimented with numerous algorithms to process the sensor data. Examples include conventional thresholding algorithms that use angular velocity to detect gait phases [64], musculoskeletal models that estimate intention using EMG [65], and a wide range of machine and deep learning techniques.

These gait analysis techniques have been reviewed, compared, and analysed in sev-

eral published works in the literature in recent years. Reviews have been conducted to evaluate the advantages and disadvantages of multiple sensors used for gait phase detection, taking into account factors such as the number of phases to be detected, sensor placement location, and the computational algorithms used to process the sensor readings [66]. Wearable sensors have also been compared to conventional laboratory systems for analysing gait, exploring their potential as substitutes for these traditional systems [67]. Literature reviews have covered parameters of gait, machine learning algorithms, and challenges associated with gait analysis in clinical and non-clinical applications [63]. Some authors focused on reviewing a category of computational algorithms for gait analysis, such as intelligent predictive systems [68], or deep learning algorithms [20], while others examined a specific category of algorithms and sensors, such as artificial intelligence using inertial sensors [69].

Despite their recent publication between the years 2016 to 2020, there is a lack of literature solely focused on gait analysis techniques for the high-level control of lower limb exoskeletons and orthoses. Prior to 2002, the number of papers published on exoskeletons where less than 30 per year, with a cumulative total of fewer than 500 papers. However, by mid-2019, the cumulative number of papers was approaching 4000 [70]. Given this exponential increase in research, there is a need to have a systematic review specifically focused on gait analysis for controlling lower limb robotics, utilising intelligent algorithms and techniques in particular.

Machine learning (ML) algorithms offer multiple advantages that make their use preferable over conventional gait analysis methods. Gait is temporal and the relationship between its parameters is non-linear. ML algorithms excel at mapping non-linear relationships between inputs and outputs, making them well-suited for gait analysis [71]. Furthermore, compared to conventional methods and heuristics used in analysing gait, ML algorithms are better equipped at handling data variability. This is particularly important when analysing pathological gait which exhibits high levels of inter and intrasubject variability [25]. ML implementations have demonstrated higher accuracy in predictions [25,72,73], resulting in reduced torque prediction errors [74]. They have also demonstrated lower prediction time errors [25]. ML algorithms exhibit adaptability, as they were able to form predictions under dynamic speeds [74]. Furthermore, they

shorten the time required for tuning controller parameters compared to manual tuning. Tuning the parameters of an exoskeleton controller for different patients is necessary to accommodate for variations in their gait trajectories and strength capabilities, leading to more effective rehabilitation [75]. Some ML algorithms eliminate the need for hand-crafted, or expert-selected features, as they can learn relevant features from the data [76].

The numerous advantages of ML algorithms and their increasing use were the motivation behind conducting this systematic review. This chapter will focus on research that utilised intelligent algorithms such as machine and deep learning, for gait analysis in the context of active lower limb exoskeleton and orthoses control. This chapter will explore the various gait parameters and features that researchers are detecting and utilising for these robotic devices, as well as the sensors they incorporate into their designs. This chapter is organised into nine sections. Section 2.3 outlines the research methodology adopted, explaining the criteria used to include or exclude papers from this analysis. Section 2.4 will present background information on the topics covered in the review. Sections 2.5 to 2.8 offer a comprehensive review of the selected papers, with each section focusing on a single gait parameter being detected/predicted. These sections are further grouped based on the type of intelligent algorithm utilised. Finally, Section 2.9 consists of a discussion and conclusion, where the findings of this review are analysed and summarised.

2.3 Methods

2.3.1 Research Identification

A systematic review has been conducted, which is a systemic approach to exploring existing literature. It involves selecting a set of keywords, along with Boolean operators, to extract the most relevant papers from the literature. The focus of the review is on literature that implemented intelligent machine learning algorithms for gait analysis to be used with lower limb exoskeletons and orthoses. Therefore, the chosen keywords should reflect the topic. They should be general enough to capture applicable literature

and encompass the various techniques, sensors, and terminology used by researchers in their studies, while also being restrictive to eliminate irrelevant research. The chosen keywords are: (exoskeleton OR orthosis OR orthotic) AND (gait OR locomotion) AND (recognition OR classification OR prediction OR intention OR selection OR detection OR discrimination OR partitioning OR segmentation OR estimation) AND ("machine learning" OR "deep learning" OR "artificial intelligence" OR "neural" OR "fuzzy").

2.3.2 Databases for Research Extraction

The aforementioned keywords were used to retrieve papers from two databases: IEEE and SCOPUS. Initially, PubMed was also included but no relevant papers were found.

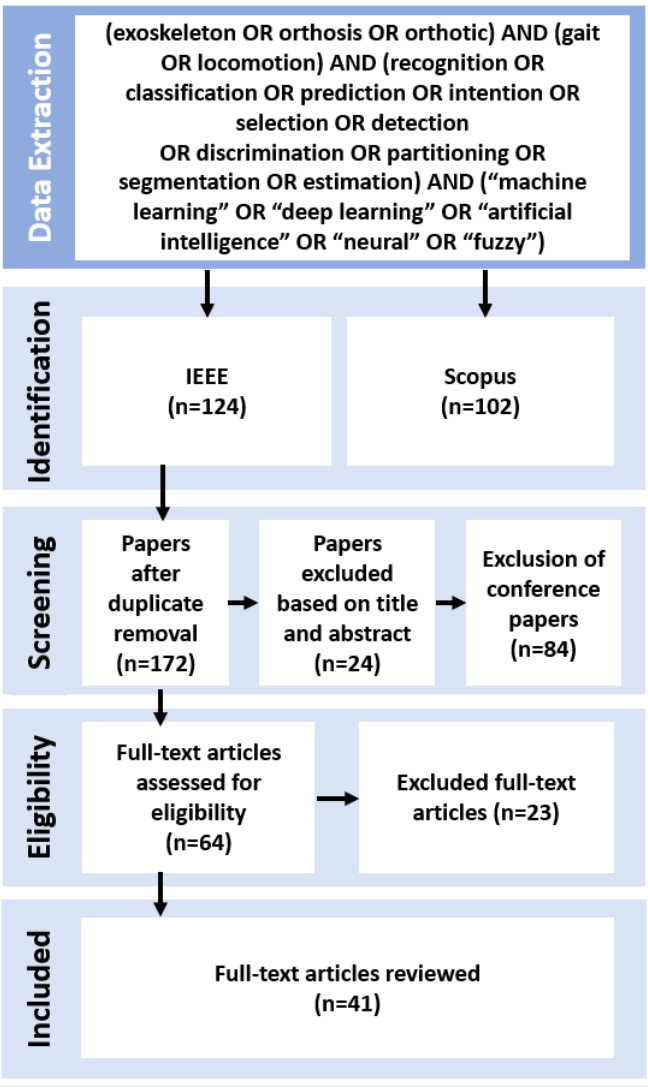

Figure 2.1: Flowchart on the methodology of article selection.

2.3.3 Inclusion and Exclusion Criteria

In addition to the keywords, specific inclusion and exclusion criteria were applied to refine the results. The review includes research published between the years 1989 to May 2020 in the English language. Results were limited to journal and conference papers only. There were a few studies on SCOPUS where the full-text papers were not available or inaccessible and hence were not included. A total of 226 papers met the inclusion and exclusion criteria. Duplicate papers available on both databases have been removed, reducing the papers to 172. These papers were initially analysed based on their abstract, and irrelevant or less relevant papers were manually removed. Reasons for exclusion include the use of prostheses rather than orthoses or exoskeletons, focus on the upper limb rather than lower limb robotics, and the absence of intelligent machine or deep learning algorithms. Results were further limited to research articles, excluding conference papers. Out of the 64 full-text articles assessed for eligibility, 41 were included in this review. The full texts of these articles were reviewed, focusing on the parameters the researchers considered, the intelligent algorithms used, the sensing modalities employed, the types of subjects used for testing or training the algorithms and the overall system performance. This process is visually illustrated in Figure 2.1.

2.4 Gait Parameters

According to Whittle, the term 'gait' is a technical terminology used to describe 'the manner or style' we walk in [77]. When studying or describing gait, multiple parameters are observed. These parameters typically have normal ranges for healthy gait, with variations due to several factors such as anthropometric parameters (i.e. age, height and limb lengths) [78]. Pathological gait parameters often deviate from these typical ranges. In this section of the chapter, four gait parameters will be introduced: (1) gait phase, (2) locomotion mode, (3) trajectory and joint angle, and (4) torque and moment. Those gait parameters have been detected, predicted and analysed for the control of lower limb robotics using different mechanisms and techniques. Refer to Appendix A for more details on the machine learning algorithms used for the detection/prediction

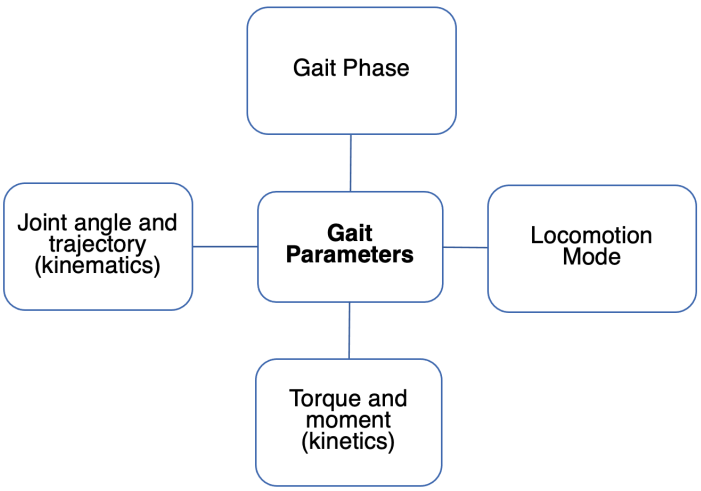

Figure 2.2: The four categories of gait parameters that have been predicted/detected in the reviewed studies for the control of lower limb exoskeletons.

of those gait parameters.

2.4.1 Gait Phase

There are cycles of events that periodically repeat during gait. Each cycle includes a stance phase, during which the lower limb is in contact with the ground, and a swing phase where there is no contact. The stance phase can be divided into four periods: (1) loading response, which starts with heel strike (also known as initial contact), (2) midstance, when the foot is flat on the ground due to a dorsiflexion moment, (3) terminal stance, when the heel begins lifting from the ground, and (4) pre-swing, which is the last period of ground contact before the foot is lifted into the swing phase. The swing phase is further divided into three more periods: (1) initial swing, (2) mid-swing and (3) terminal swing. Overall, there are seven phases in a gait cycle (see Figure 2.3), and this sequence of events alternates between the right and left foot, resulting in a forward movement. During a single gait cycle, there will be periods of double support, when both legs are in contact with the ground, and single support, where one leg is in contact only [78]. Gait phases are part of the spatiotemporal parameters of gait [63].

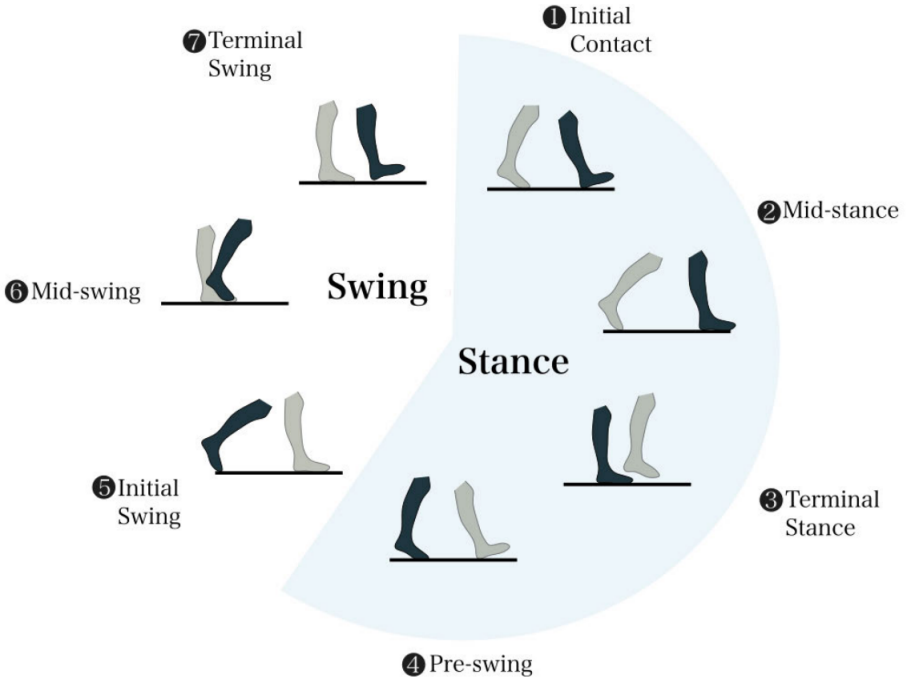

Figure 2.3: Illustration of the seven phases of the gait cycle that occur in the stance and swing periods.

2.4.2 Joint Angle

During the gait cycle, the angles of the hip, knee and ankle joints undergo periodic changes. Their values are often measured in the sagittal plane, where the greatest movement is observed, compared to frontal and transverse planes [78]. Joint kinematics [63], in addition to their first- and second-time derivatives, namely angular velocity and angular acceleration, are commonly measured and reported in gait analysis.

2.4.3 Torque/Moment

The moment of force refers to the rotational effect caused by the application of force. The magnitude of the moment depends on both the magnitude of the force applied and the shortest distance between the location of the force's application and a fulcrum or pivot. This distance is referred to as the lever arm. Moment and torque are terms that are often used interchangeably but there is a slight difference between moment and torque. Moment results in bending and torque in rotating and twisting [79]. In the context of biomechanics, a moment of force happens when the muscles contract, causing the knee joint, which serves as the pivot, to rotate. Moments can be internal

or external. Internal moments result from tension in soft tissue or from the contraction of muscles (eccentric, concentric, and isometric contraction), while external moments are due to external forces such as gravitational force. The rotational impact of these moments on joints is the cumulative sum of all the individual internal and external moments [78]. Moment of force is a kinetic parameter of gait [63].

2.4.4 Locomotion Mode

Locomotion modes encompass various physical activities including ground-level walking, standing up, sitting down, ascending and descending a slope, and ascending and descending a staircase.

2.4.5 Other Parameters

In addition to the four gait parameters introduced above, the term 'intention' has been used in some of the reviewed papers. 'Intention' in the field of human-robot interaction has been defined as "the need for the robot to have knowledge of some aspect of the human's planned action in order for the robot to appropriately assist toward achieving that action" [18]. The measurement of intention can occur at the central nervous system level based on the brain's electrical activity, or at the peripheral nervous system level based on the muscle's electrical activity, or based on interaction forces between the human and robot which can be measured with force sensors. Intention can be characterised by discrete states, which can initiate certain movements or control mode transitions, or by continuous states such as the desired position trajectories [18].

The two most common neural correlates reported by researchers for measuring intention from the brain are Movement-Related Cortical Potentials (MRCP) and Event-Related Desynchronization (ERD) [80, 81]. These neural correlates can be measured using electromyography (EEG). MRCP has a component called Bereitschaftspotential (BP), which begins around two seconds before the start of voluntary movement [82]. Meanwhile, significant ERD was observed to happen one second before movement initiation and involves a reduction in spectral power of the alpha and beta bands, which have

frequencies in the ranges of 8 to 13 Hz and 14 to 30 Hz, respectively [83]. Two other neural features mentioned in He et al's review paper on existing brain-machine interfaces used for lower-limb exoskeletons and orthoses control are the rate of neuronal firing and Steady-State Visual Evoked Potentials (SSVEPs) [84].

Muscles are effectors that execute commands from the central nervous system, contracting as a result. The electrical activity of muscles can be measured using surface electrodes, which allows the detection of voluntary activity initiation shortly before joint movement. The time delay between the activation of a muscle and the production of force is known as electromechanical delay [85].

2.5 Gait Phase

The magnitude of assistive torque provided by the active exoskeleton or orthosis may vary according to the phase of gait [74], making the detection of some or all gait phases essential. A wide variety of wearable sensors and algorithms have been utilised for this purpose. The granularity of the phase detection varied, ranging from as low as two phases up to eight phases. This section will include a list of machine learning algorithms used for gait phase detection. The identified phases and the overall system performance have been discussed in detail. Refer to Appendix B.1 for a table that summarises the details of all the papers that predict gait phases.

2.5.1 Neural Networks

Jung et al [72] implemented a multilayer perceptron neural network (MLPNN), to detect two gait phases: stance and swing. Their study focused on controlling ROBIN-H1, an exoskeleton designed for the rehabilitation of stroke patients. Pitch orientation and angular velocities were measured by several sensors and were used as input to the MLPNN. Ground truth labels were obtained using force plates. The MLPNN had one hidden layer with growing nodes that increased from 5 to 50. The back-propagation algorithm was used for optimisation. Classification success rate (CSR) was the metric used to evaluate performance, which accounted for three types of errors:

early classification, delayed classification and erroneous classification. The authors considered erroneous classification errors the most dangerous type of error for robotic control and a potential risk of injury. Both offline and online tests were conducted based on data from healthy subjects. The average CSR rate for the offline test was 97.75%, which was higher than the average CSR for the online test, which was 90.75%.

Ma et al [86] also implemented an MLPNN to detect four phases of gait: heel strike, flat foot, heel off, and swing. It was compared to the kernel recursive mean square method (KRLS). The average classification rates of the MLPNN for 3, 5, and 10-fold cross-validation were 83.17%, 82.42%, and 83.23%, respectively. However, the MLPNN showed lower performance compared to KRLS, with a difference of 2.33%, 3.62%, and 3.04% in classification rates for the respective classification cross-validation folds.

Kang et al [74] implemented a neural network to estimate the percentage of the gait cycle for controlling a bilateral hip exoskeleton. Instead of discrete phases, the gait was considered as a continuous variable, and the model aimed to detect the percentage of the gait phase. Different combinations of sensors were experimented with, and a neural network with one hidden layer and 20 nodes was trained. The authors found that using all sensors combined resulted in greater error than using the hip encoder and thigh IMU only, possibly due to the simple architecture of the model used. Three models were evaluated: a generalized/independent model (trained on data from multiple users and tested on data from an unseen user), a user-specific/dependent model (trained on data from a single user and tested on unseen samples from that user), and a semi-dependent model (trained on data from multiple users and tested on unseen samples from one of the users it initially trained on). These models were compared to a time-based estimation (TBE) model that uses FSRs, under varying gait speeds. For steady-state speeds, there wasn't a significant difference in performance, and the generalized model had the highest error even compared to TBE. However, under dynamic and extrapolated speed dynamic movements, the user-specific and semi-dependent models showed higher accuracies compared to TBE, reducing estimation error by 23.4% and 26.3% respectively. The enhanced estimation led to a decrease in torque generation error by 32.4% ($p < 0.05$) for the dependent model and by 40.9% ($p < 0.05$) for the semi-dependent model. The semi-dependent model showed the best performance.

Hua et al [87] implemented an adaptive neural-fuzzy inference system (ANFIS), which combines Takagi-Sugeno fuzzy inference system and neural networks, for detecting two phases: stance and swing. Their objective was to develop a lower limb exoskeleton capable of withstanding the lifting of heavy loads. Plantar pressure readings and their first and second derivatives were used as input for predictions. A trapezoidal membership function was employed. The authors demonstrated the model's generalization potential by segmenting gait phases under various locomotion modes.

Nazmi et al [88] implemented an MLPNN to detect stance and swing using EMG signals. Two types of optimisation algorithms, the scaled conjugate gradient (SCG) algorithm and the Levenberg-Marquardt (LM) algorithm were compared. The SCG achieved optimisation in a shorter duration, while the LM resulted in higher accuracy and lower MSE. Numerous features derived from the muscle activity were used for training the neural network including mean absolute values (MAV) only, mean absolute value and waveform length (group 1), and mean absolute values, waveform length, RMS, SD and integrated EMG (group 2). Using the LM optimization algorithm, the accuracy using MAV features was 78.6%, for group 1 features was 82.3%, and for group 2 features was 87.4%. These results indicated that using the greatest number of features leads to more accurate predictions.

Zhang et al [89] implemented a backpropagation neural network (BPNN) to detect five phases of gait. The authors investigated the effect of variations in the load carried by the users on the EMG activity and the ability of an algorithm trained on data from users carrying one load level to perform accurate predictions when tested on data from users carrying multi-load levels. EMG data was collected from users carrying multiple loads as a percentage of their masses (0%, 20%, 30%, and 40%) while walking on a treadmill at three different speeds. The authors found out that intra-load testing, which involves training and testing a model using data from users carrying one load level, had higher accuracy than inter-load testing, which involves training a model using data from users carrying one load level and testing it using data from users carrying another load level. These findings indicate that muscle activity changes with load variations, and emphasises the importance of training the algorithm with data from multi-load conditions to maintain the performance of the exoskeleton in various conditions.

2.5.2 Deep Neural Networks

Zhen et al [90] implemented a long short-term memory and deep neural network (LSTM-DNN) for the detection of two phases: stance and swing, using acceleration signals obtained from an inertial measurement unit. The LSTM had 36 units, and its output was fed into a DNN. The LSTM-DNN was tested under three predefined speeds, achieving accuracies greater than 91.8% and F-scores greater than 92%. The LSTM model on its own had lower performance, with accuracies greater than 86.8% and F-scores greater than 86.4%.

Wang et al [33] implemented a deep memory convolutional neural network (DM-CNN) to detect four phases of gait, using foot pressure sensors and IMUs. Phase classification using the single-mode (i.e., training the model on data from a single sensor) and multi-modal approach (i.e., training the model on data from multiple sensors) were compared. The multi-modal achieved a higher performance, and classification with acceleration signals alone outperformed classification using pressure signals alone.

Jung et al [72] implemented a non-linear autoregressive with external inputs (NARX), which is a type of recurrent neural network. The performance of NARX was slightly lower than the MLPNN, with an offline CSR of 97.05%, and an online CSR of 91.93%. However, NARX had fewer unstable regions, which are oscillations in the output, compared to the MLPNN. The authors suggested that segmenting the data into individual strides for training may have resulted in discontinuities in the gait pattern, affecting the performance of NARX.

2.5.3 Decision Trees/Random Forest

Farah et al [91] implemented a logistic model decision tree (LMT) for the detection of four gait phases: loading response, swing, terminal swing, and push-off using knee angles, thigh angular velocity, and acceleration as input features. The chosen LMT model was a J-48 decision tree with terminal nodes performing a logistic function. The C4.5 decision tree splitting criteria was followed, resulting in a tree with a total size of 1643 and 822 nodes. To remove erroneous or misclassifications and improve

accuracy, a transition sequence validation and correction algorithm (TSVC) was applied post-classification. The training and validation accuracies were 98.76% and 98.61% respectively. The F-score for the validation set was 0.97. It was noted that a large proportion of false negatives were due to transition periods during the phases.

Pasinetti et al [92] implemented a random forest (RF) algorithm to detect two phases of gait: stance and swing, using time of flight cameras. Depth cameras were embedded in two crutches used when walking with the exoskeleton, with each camera monitoring the contralateral leg. The images captured were processed to separate the leg and floor from the environment. A plane detection algorithm identified the floor's surface as a reference for measuring distances between it and other objects, i.e., foot. The percentiles of these distances were used as features for classification. Two variations of algorithms were implemented, a random forest (RF) comprised of decision trees and a sigma-z random forest. The Sigma-z RF accounts for uncertainties such as measurement errors, and for variances in the data that can occur if there is similarity or overlap between features of distinct classes. Sigma-z outputs a classification, in addition to a confidence value for that classification. The choice of admittance threshold influenced a trade-off between classification accuracy and the number of unclassifiable samples. The RF and sigma-z RF achieved accuracy values of 81% and 87.3%.

2.5.4 Fuzzy Logic

Chinimilli et al [93] implemented fuzzy logic to detect four gait phases, with the aim of creating an adaptive virtual impedance controller for a knee exoskeleton. Ground contact forces measured with smart shoes were input to a fuzzy logic algorithm, which consisted of partial trapezoid and triangular membership functions. The classified gait phase, along with the locomotion mode (further discussed in the following section of this chapter), were used as inputs to a Gaussian mixture model (GMM) that determined the appropriate stiffness and damping ratio for controlling the exoskeleton. While the classifier's performance was not recorded, automatic impedance tuning that is based on the outputted gait phase and locomotion mode has been assessed in comparison to constant impedance and finite state machines. The automatic impedance control

showed desirable results, decreasing the EMG activity level (of the vastus medial), shortening the step length and increasing the cadence compared to the two other control modes.

Chen et al [94] implemented fuzzy logic for the detection of four phases using foot pressure data. The HEXO exoskeleton utilised hybrid control, with adaptive impedance control (AIC) in the stance phase and active-disturbance rejection control with fast terminal sliding mode control (ADRC-FTSMC) in the swing phase. The fuzzy logic algorithm for phase detection used a sigmoid membership function.

Huo et al [95] used fuzzy logic for gait phase detection for E-ROWA, an exoskeleton that supports walking by generating appropriate torque according to the phase of gait. The authors also detect locomotion modes, discussed in the following section.

2.5.5 K-nearest Neighbour

Chen et al [96] implemented the k-nearest neighbour (KNN) method to detect eight gait phases using joint angle sensors that measure the angle of the hip, knee, and ankle joints, and plantar pressure sensors.

Wang et al [33] implemented a KNN algorithm for the detection of four gait phases. However, KNN's performance was lower compared to DM-CNN, N-HMM, and HMM, with an accuracy of 88.5%, 8.6% lower than the best-performing algorithm. Its precision and recall values were 81.5% and 82.5% respectively.

Zhen et al [90] implemented a KNN algorithm for the detection of two gait phases. Seven different values of K were used, ranging from 2 to 30, and the Euclidean distance was set as the distance parameter. The algorithm's performance was evaluated at three walking speeds. The KNN had the worst performance compared to LSTM-DNN, LSTM, and SVM. Its accuracy ranged between 69% and 76% while its F-score was between 70% and 77%.

A KNN is implemented by Zhang et al [89], for the detection of five gait phases. The KNN algorithm had a lower performance than BPNN.

2.5.6 Hidden Markov Model

Manchola et al [73] implemented a hidden markov model (HMM) to detect four gait phases using a single IMU placed on the instep of the foot. Gyroscope signals in the sagittal plane were used instead of acceleration signals that require a Kalman filter to address the issue of drift. The Baum-Welch algorithm was used for model training and the Viterbi algorithm for gait phase recognition. Two types of training have been performed: inter-subject/subject-specific training (SST), where the algorithm was trained on data from a single user and tested on unseen samples from that user, and inter-subject/standardised parameter training (SPT), where the algorithm was trained on healthy users and tested on unseen samples from one of those healthy users or on a patient. SST had higher accuracy compared to SPT, and the difference was more pronounced for healthy subjects than for patients. The SST accuracy was 81.44% for healthy users and 78.06% for patients, meanwhile, the SPT accuracy was 76.91% for healthy users and 76.36% for patients. Despite the training approach, the HMM accuracies were still higher than the threshold-based algorithm, which was a finite state machine that classifies gait by detecting peaks, troughs and zero-crossings of the signals of an accelerometer and gyroscope.

Wang et al [33] implemented both HMM and N-HMM as a comparison to DM-CNN. However, both had lower accuracies compared to DM-CNN model, scoring 96.2% and 92.3% respectively.

2.5.7 Support Vector Machines

Ma et al [86] implemented a support vector machine (SVM) to detect four gait phases and compared its performance to the kernel recursive least-square method (KRLS) and MLPNN. The SVM used a Gaussian kernel function and was optimised using particle swarm optimization. The SVM achieved accuracies of 83.00%, 82.69% and 83.29% for 3, 5, and 10-fold cross-validation, respectively. However, its performance was lower than that of the KRLS method.

Zhen et al [90] implemented an SVM for the detection of two gait phases. Four kernel

functions were used: linear, RBF, sigmoid and poly with the RBF kernel resulting in the highest performance. The SVM achieved slightly higher performance than the KNN but was outperformed by the LSTM-DNN and LSTM models.

Zhang et al [89] implemented an SVM and compared its performance to BPNN, KNN, and SVM. The SVM and kNN demonstrated lower performance than the BPNN.

2.5.8 Principal Component Analysis

Tanghe et al [97] implemented a probabilistic model, specifically the probabilistic principle component analysis (PPCA), for the prediction of four phases of gait before they occur, which are initial contact, flat foot, heel off, and toe off events along with joint trajectories. The training dataset used consisted of data from healthy participants walking without an exoskeleton. The models were validated on two separate data sets: one with healthy participants walking without an exoskeleton and another with healthy participants walking with an exoskeleton. In this paper, zero error signifies that the phase was predicted 0.2 seconds before to the actual event. The maximum median error for the exoskeleton-free validation dataset across all four phases was 9ms. However, when considering the dataset involving the exoskeleton, the median was higher, reaching 15ms for initial contact, and 33ms for toe-off. It was also noted that heel-off was the most challenging phase to predict.

2.5.9 Kernel Recursive Least Square Method

Ma et al [86] implemented a kernel recursive least square method (KRLS) for gait phase detection, using the knee and hip joints angle as input. The reported accuracies for gait phase classification were 85.49% for 3-fold cross-validation, 86.04% for 5-fold cross-validation, and 86.26% for 10-fold cross-validation.

2.6 Locomotion Mode

Identifying the locomotion mode is essential for robotic devices that provide assistance with daily life activities. This is because each activity requires specific assistive requirements, and identifying the mode allows for smooth transitions between the different activities. Locomotion modes can be classified into static or dynamic modes. The main static locomotion modes are sitting and standing. The main dynamic locomotion modes are straight-level walking, ascending stairs, descending stairs, ascending slope, and descending slope. Some authors also focus on detecting mode transitions, such as sitting to standing or level walking to ascending etc. Often, the identification of locomotion mode is performed in conjunction with gait phase detection, either before or after identifying the current phase. Refer to Appendix B.2 for a table that summarises the details of all the papers that detect locomotion modes.

2.6.1 Neural Networks

A BPNN is implemented by Song et al [98], for locomotion mode detection. They identified 4 static, and 11 dynamic modes, resulting in a total of 15 locomotion modes including sitting, standing, level walking, and level walking with weight, among others. IMUs and foot pressure sensors were used to acquire signals, from which they extracted time domain, frequency domain and energy features. A total of 141 features have been extracted, including mean, variance, correlation coefficient, wavelet energy entropy, SMA, Fourier series and maximum values. Some features are more suited for classifying static modes such as leg angles, while others were more appropriate for dynamic modes. Three neural networks were developed, each with three layers. The first network determined whether the mode is static or dynamic and contained 5 input, 25 hidden, and 1 output node. Depending on the outcome, a static mode or a dynamic mode neural network was employed. The static neural network has 20 input nodes, 100 hidden nodes and 1 output node, while the dynamic had 40 input nodes, 200 hidden nodes and 1 output node. The authors found it easier to classify dynamic modes due to their larger inherent pattern differences compared to static modes. However, they

observed some confusion between standing and sitting with a load. For single-mode classification, the overall accuracy was 98.28%. The accuracy of multi-mode classification, which involved transitioning between modes, was lower.

An ANN is implemented by Islam et al [99] to detect: level walking, ascending stairs/ramp, and descending stairs/ramp, for controlling their Portable Powered Ankle-Foot Orthosis (PPAFO) using IMU and foot pressure sensors. Calibration was performed with every step to address drift, particularly at the zero-acceleration stage (mid-stance). Foot pressure sensors assisted in identifying mid-stance for calibration. The input of a three-layer neural network consisted of six tapped delays of vertical foot velocity and angle measurements derived from IMU. The hidden layer had 10 nodes and the output layer had 3 nodes. Subject-specific training was conducted, and the accuracies ranged from 97.8% and 100%. The authors also measured the time required to detect a transition as a percentage of the gait cycle, representing how much of the gait cycle elapsed before a mode transition is detected. Mode transitions were identified in the swing phase of the transitioning step, within 28% of the gait cycle for the stair mode and 16% of the gait cycle for the ramp mode.

Backpropagation neural network (BPNN) and radial basis function neural network (RBFNN) are implemented by Wang et al [100], to detect six locomotion modes with IMUs and plantar pressure sensors. Both networks had an architecture consisting of 20 nodes for input, 12 nodes for the hidden layer and 6 outputs. The BPNN outperformed the RBFNN, achieving an accuracy of 93.3% compared to 91.2%.

2.6.2 Support Vector Machine

Wang et al [100] implemented an SVM to detect six locomotion modes. The SVM outperformed two algorithms, the Radial Basis Function Neural Network (RBFNN) and Backpropagation Neural Network (BPNN), achieving an accuracy of 96.5%. When comparing the choice of kernel functions, the linear kernel resulted in better performance compared to the polynomial function.

Villa-Parra et al [101] implemented an SVM with a Gaussian kernel for locomotion

intention prediction based on EMG. The authors used muscle activity from the trunk and compared it to that of the lower leg to predict the intention to perform various locomotion modes such as flexion-extension of the knee, standing up, sitting down etc. The reported accuracies ranged between 76%-83% and 71%-77% for lower limb and trunk muscles. The comparison between trunk muscles and lower limb muscles aimed to assess their abilities to accurately predict locomotion intentions. The goal was to determine if the truck muscles can be used as an alternative to lower limb muscles which are often affected by pathologies or weakness in patients with preserved trunk muscle activity.

Goh et al [76] implemented an SVM as a comparison to a Spectral Representation Learning Model (SSRL) for the detection of four locomotion modes with EEG. Principle Component Analysis (PCA) and F-score (FS) were used for dimensionality reduction. SVM-PCA and SVM-FS performed worse than SSRL, although SVM-FS outperformed SVM-PCA.

2.6.3 Deep Neural Networks

Hua et al [87] implemented a deep neural network (DNN) and convolutional neural network (CNN) for the detection of 6 locomotion modes. They have experimented with several machine learning algorithms, including DT, DA, KNN, SVM, EM which were pre-processed with kPCA to reduce the dimensions of the input features. The deep learning algorithms outperformed traditional ML algorithms achieving 52% higher efficiency, without the need for kPCA, with a lower duration (20ms) to perform the computation. The stacked autoencoder DNNs, optimized with Genetic Algorithm Particle Swarm Optimization (GA-PSO), achieved accuracies around 99%, compared to 97.2% prior to optimization.

Goh et al [76] implemented a spatio-spectral representation learning model (SSRL) to detect four locomotion modes using EEG signals. The locomotion modes include level walking without an exoskeleton, and level walking with an exoskeleton at zero, low, and high torque support modes. The SSRL consisted of three hidden layers: a spatial layer, a spectral layer, and a fully connected layer. EEG data from 27 healthy male subjects

was collected from 20 channels. The models were trained using wide spectral frequencies (WS) ranging from 1-42 Hz or prominent spectral frequencies (PS) ranging from 8-30 Hz. SSRL was compared with two other machine learning algorithms, random forest and support vector machines with radial basis function kernels, that had dimensionality reduction methods, including principal component analysis (PCA) and F-score (FS) applied to them. The SSRL-WS had the highest accuracy, 77.8 ± 1.8%. The accuracies of SVM-FS, SVM-PCA and RF-FS algorithms for the wide spectral frequencies were 74.3 ± 1.6%, 64.3 ± 1.8%, and 65.9 ± 1.8%, respectively. The accuracies when using the prominent spectral frequencies were lower, but the algorithms still had the same order of performance, with SSRL achieving 72.9 ± 1.7%, while SVM-FS, SVM-PCA and RF-FS achieved 70.2 ± 1.6%, 54.8 ± 2.0%, and 58.9 ± 1.7% respectively. The results showed that FS performed better than PCA, but the SSRL, which obviated the need for handcrafted features, still performed better than both. Additionally, the spatial parameters of the network provided insights into the role of different brain regions in controlling gait, contributing to the understanding of the topographic organization of the brain.

2.6.4 Fuzzy Logic

Chinimilli et al [93], implemented a fuzzy inference algorithm to detect three locomotion modes: level walking, uphill and downhill walking. The algorithm utilised the detected heel strike (right leg) and the knee angle derived from IMUs placed on the thigh and shank. The number of locomotion modes detected was limited by the experimental setup, which involved a treadmill only.

Parri et al [102] implemented a fuzzy logic algorithm to detect seven locomotion modes for the control of a hip orthosis. The algorithm first used a threshold-based algorithm to categorise whether a mode is static or dynamic using hip angles. Static modes including sitting, standing and transitioning between the two were identified with this approach. Dynamic modes, including level walking, ascending and descending stairs were classified using the fuzzy logic algorithm based on hip joint angles and centre of pressure measured by sensitive insoles. A Gaussian function was used as a membership

function. The authors aimed to create a generalised model and avoid subject-specific training, therefore, membership values were based on data from 6 subjects with varying speeds and assist modes. The effect of inter-subject variability had a noticeable impact on the reduction of performance for a participant who was taller than the others.

Huo et al [95] implemented a fuzzy logic algorithm to detect five modes of gait. The Mamdani fuzzy inference system has been specifically employed with bell-shaped curves as membership functions. Four healthy participants wearing the E-ROWA exoskeleton were asked to walk under normal and simulated abnormal gait, with the abnormal gait achieved by locking the flexion of one knee joint. The algorithm achieved accuracies greater than 97.7% for normal gait and 97% for abnormal gait. The authors also reported the latency of detection, with modes detected at the start of the step rather than the end, was less than $32 \pm 8.3\%$ of a step.

2.6.5 Random Forest/Decision Trees

Novak et al [103] implemented a Decision tree algorithm to decode the intention of gait initiation and termination, using IMU and pressure insoles. Gait initiation was divided into two parts: onset which includes events that occur in preparation for toe-off, approximately 0.5 s before the foot begins to lift from the ground, and toe-off which is when the foot is fully lifted from the ground. Gait termination is when a person decides to stop walking. Within-subject trials yielded better results than subject-independent trials. The use of IMU and pressure sensors separately and combined as input to decision trees have also been compared. IMUs performed better in detecting gait termination, however, pressure insoles were still needed particularly for the segmenting gait cycles. In 80% of within-subject classification trials, gait termination was predicted prior to the actual event. The research aims to assist a user with their first step upon detection of their intention to walk or terminate assistance when they desire to stop walking. The authors noted that the effects of an assistive device, such as an active orthosis, on 'masking' intention are yet to be evaluated.

Goh et al [76] implemented a random forest (RF) algorithm to detect four locomotion modes with EEG. Pre-processing involved dimensionality reduction using F-score (FS).

RF-FS algorithm performed worse than the SSRL model to which it was compared.

2.6.6 Multiple Kernel Learning

Zhang et al [26] implemented multiple kernel learning (MKL) algorithms to decode intention for performing four locomotion modes using EEG including forward walking, stopping, turning left and turning right. 64 channels of electrodes were used and placed according to the 10-20 international system. The brain was segmented into 13 regions, and the algorithm's performance and role of the brain regions were evaluated. The frontal and front-central regions, particularly, the MFC and RFC had the highest weighting indicating the largest role in intention/limb control. However, the exact precedence of those regions differed in the healthy and SCI participants, with the healthy participant having the highest weighting in MFC followed by the RCF region, while the SCI participant had the reverse order. The accuracy of the model was 74.5% for the healthy participant and 68.4% for the SCI participant in the single-session experiment. In a nine-session experiment over 30 days, the classifier was set to detect walking and stopping only with an accuracy greater than 90% and resulted in a change in the weighting of the regions, indicating cortical plasticity. The weighting and accuracy improved as sessions progressed, and the stopping mode was the most difficult to detect among the four modes.

2.6.7 Sparse Discriminant Analysis

Gui et al [27] implemented linear discriminant analysis (LDA) to detect intention for four locomotion modes: stopping, level walking, accelerating, and decelerating. The authors used cognitive and peripheral signals, with LDA relying on brain-generated Steady-State Visual Evoked Potentials (SSVEP) as input features. The central pattern generator uses the output of the LDA to generate the exoskeleton's trajectory. The recognition rate of steady-state (ROS) was 92.40%, and the duration of transient state (DOT) or time delay was 1.7 seconds. The authors discussed the impact of increasing the locomotion modes from four to eight on the quality of SSVEP, which reduced the recognition rate to 70%. EEG was used for transitioning between the four discrete

modes, whereas continuous locomotion speed control relied on EMG. EMG was the input to an admittance model with further details discussed in their paper.

Lopez-Larraz et al [28] implemented sparse discriminant analysis (SDA) to identify gait initiation intention for an exoskeleton that assists patients with incomplete spinal cord injuries. From EEG signals that were generated and cue-guided, event-related desynchronization (ERD) and movement-related cortical potential (MRCP) features were derived and used with SDA. The model was tested on data from three healthy participants and four SCI patients. The experimental protocol contained four parts: rest, preparation, attempt movement, and movement. The accuracy for decoding the intention of healthy users was higher than those with SCI, with 88.44 $\pm$ 14.56% compared to 77.61 $\pm$ 14.72%. Frequency ERD features were more commonly chosen over temporal MRCP features.

2.6.8 Canonical Correlation Learning

Zheng et al [104] implemented the canonical correlations algorithm (CCA), an unsupervised learning algorithm, to decode intention for three motion patterns. The authors used steady-state visual evoked potential (SSVEP) features. They conducted both offline and online experiments to evaluate the performance of their approach. In offline experiments, visual stimuli were presented to participants, followed by intention classification based on their EEG. In online experiments, participants were exposed to visual stimuli and then performed physical motions including squatting, walking then standing. This setup involved multi-modal data as input features, including foot pressure and joint positions. In both experimental scenarios, the authors achieved a classification accuracy of over 90%.

2.7 Moment/Torque

Torque and moment are kinetic parameters of gait that have been detected using various machine learning algorithms as follows. Refer to Appendix B.3 for a table that summarises the details of all the papers that predict joint torque/moment.

2.7.1 Neural Network

Ma et al [86] implemented an MLPNN to predict hip joint assist torque to support the extension and flexion of the hip. For hip extension assistance, torque needs to be provided in heel strike and flat foot phases. For hip flexion assistance, torque needs to be provided in the heel-off and swing phases. The authors developed a generalised model that was trained on one group and tested on another. However, when compared to the kernel recursive least-square method, the MLPNN's mean square errors were twice as large.

Gui et al [105] implemented a radial basis function neural network (RBFNN) to output passive and active torques. This method serves as an alternative to the hill-type model, often used for deriving torque from EMG activity. The torques generated by the RBFNN are utilised by a motion controller, the extended Slotine-Li, that controls an assist-as-needed exoskeleton. Two neural networks are employed in this setup; one uses motion states as input and outputs passive torque, while the other uses motion states and processed EMG signals as input and outputs active torque. The first RBFNN is trained with the muscles relaxed until a satisfactory steady-state error level is reached. Subsequently, the second network is trained while asking the subjects to perform voluntary active torque, keeping the parameters of the first network constant. The outputs are compared against measurements from torque sensors. The authors calculated torque during the swing phase only, due to the unavailability of force plates. The utilisation of the Slotine-Li scheme eliminated the need for EMG signal calibration. To achieve 'activity-based neuroplasticity' the delay between estimated and actual torques needs to be below 300ms. This has been achieved as the time delay ranged between 174ms and 305ms. In future work, the authors suggest using high-density EMG.

Xiong et al [106] implemented a neural network for internal joint moment prediction using EMG signals and joint angles. Prior to the neural network, an elastic net was used to eliminate noise and redundant features in the input, resulting in reduced computational load and enabling real-time prediction. The neural network was trained on the data from 8 healthy subjects. The outputs of the network were compared to joint moment values calculated via inverse dynamics, which the authors acknowledged is a

source of limitation since the calculated values do not account for the presence of errors between calculated values and actual internal torques.

Xiong et al [107] implemented an extreme machine learning neural network (ELM) to predict joint moments. A hill-muscle model is used to determine the essential features for making accurate predictions. The authors successfully calculated moments of various joint movements, including flexion and extension of the hip and knee, abduction and adduction of the hip, and the plantar and dorsiflexion of the ankle. The performance of the model was evaluated using the variance accounted for (VAR). Results showed that incorporating muscle activity, muscle actuate joint angles, and angular velocities (instead of joint angles and angular velocities) yielded the best performance, with a VAF of 89.67 $\pm$ 5.56%. Conversely, using EMG alone resulted in a significantly lower VAF of 82.83%. This highlights the inadequacy of relying on EMG signals alone for joint moment calculation. Furthermore, the study demonstrated that foot pressure sensors were not needed.

2.7.2 Kernel Recursive Least-Square Method

Ma et al [86] implemented the kernel recursive least-square method (KRLS) to predict hip joint assist torque. It has been compared against MLPNN, with MLPNN performing worse than KRLS.

2.7.3 Joint Angle and Trajectory

Joint angles and trajectories are considered a kinematic parameter of gait and have been primarily predicted using neural networks and principal component analysis. Refer to Appendix B.4 for a table that summarises the details of all the papers that predict joint angles.

2.7.4 Neural Network

Kutilek et al [108] implemented an artificial neural network (ANN) to predict joint angles using cyclograms. The authors used a neural network trained with backpropagation, where the inputs included current joint angles (for a single joint), angular acceleration, weight, and age. They also trained a separate neural network that incorporates inclination angles calculated using PCA to the angles of two joints as inputs. The predictions for the hip-knee were found to be more accurate than ankle-knee angle predictions, and the neural network that utilised inclination angles performed better. In another study published a year later, Kutilek et al [109], expand on their work by including the second moment of area (x-axis and y-axis) as additional inputs for joint angle prediction.

Mazumder et al [110] implemented a radial basis function neural network (RBFNN) to generate gait trajectories. The model incorporates joint angles derived from IMU signals, gait phases and stride times calculated based on EMG signals, and foot pressure sensors as input to the RBFNN. The model was trained on data from 5 healthy subjects and is capable of adapting to the user's walking pace and anthropometrics.

Lee et al [111] implemented RBFNN and MLPNN for the calculation of joint angles, using EMG signals. Their focus was on using the joint angles of a healthy leg to predict the joint angles of a pathological leg with an orthosis or prosthetic. The RBFNN uses EMG signals from the rectus femoris to predict knee joint angles of the healthy lower limb while the MLPNN uses the output of the RBFNN to predict the hip and joint angles of the pathological lower limb. The experimental setup involved measuring joint angles during level walking, sitting and standing off a chair, with an average accuracy of 97.5% and an absolute average error rate of 0.25^{o}.

Xie et al [112] implemented a generalized regression neural network (GRNN) and BPNN for calculating joint angles. The inputs included EMG, hip joint angles, and plantar pressure. The GRNN was optimized with the golden-selection algorithm with a Gaussian kernel transfer function. Wavelet denoising was also implemented since high-frequency signals contribute to the instability of data affecting the output of the

GRNN. Meanwhile, the BPNN was trained with the Levenberg-Marquardt algorithm. The GRNN had a shorter prediction time than the BPNN, 2.38s compared to 16.029s respectively, for very similar correlation coefficient values.

Wang et al [113] implemented an Elman neural network to detect knee joint angles using EMG. The Elman network has a feedback mechanism and a memory component. EMG signals were recorded during leg extension exercises at various speeds, both with and without load. The researchers employed a multilevel wavelet decomposition and calculated the correlation dimension of wavelet coefficients (WCCD) that were used with the Elman neural network. The WCCD achieved the lowest RMSEs compared to time-domain feature extraction algorithms, IEMG and RMS, and frequency-domain feature extraction algorithms, MNP. The Elman Neural network outperformed other algorithms, including BPNN, LSSVM, GRNN. The study reported that higher speeds during the extension exercises resulted in larger RMSEs. However, when comparing exercises with and without load under constant speeds, having load led to lower RMSEs.

Gomes et al [114] implemented three MLPNNs to generate trajectories for a lower limb orthosis. The first neural network approximated inverse dynamics and calculated torque variations. The second neural network performed optimization to determine the adapted step time. The third neural network calculated the trajectory input to the orthosis's position controller including position, velocity, and acceleration values. By considering the zero-moment point criterion and interaction forces between the user's limb and the orthosis, the users can voluntarily alter their walking pattern, such as increasing their speed while maintaining stability.

Wu et al [115] implemented an autoencoder neural network (AENN) for the prediction of hip and knee joint trajectories for their lower limb robot SLEX. The generated patterns were based on walking speed and 21 body parameters. A Gaussian regression process (GRP) with automatic relevance determination (ARD) was used to map the relationship between walking speed (desired) and body parameters to spatial-temporal features. The AENN utilised the spatial-temporal features to output joint trajectories for the knee and hip angle. This approach allowed for individualized gait trajectories tailored to the person's particular body parameters, enabling the sharing of the

exoskeleton by multiple users.

2.7.5 Deep Neural Network

Boudali et al [116] implemented a recurrent neural network (RNN) for hip and knee joint trajectory prediction during transitioning locomotion modes, such as level walking and ascending stairs. The authors utilised the angular position and velocity of a cane, to predict the angular position of the contralateral foot. The RNN used for dynamic mapping was compared to the least square method used for static mapping. The RNN outperformed the least square method and demonstrated the capability of producing predictions during mode transitions (i.e., not limited to predictions during steady state), which was a limitation of the least square method. Introducing the position and velocity of an arm to the model further lowered the RMS errors of the predictions reported as 1.36^{o} and 2.48^{o} for the hip and knee joints respectively, for experiments involving intra-subject training. It was observed that the accuracy was higher for intra-subject mappings compared to inter-subject mappings.

2.7.6 Least Square Method

Ma et al [86] implemented the kernel recursive least square method for hip assist torque prediction, outperforming the MLPNN. The mean square errors were between -11.18 and -13.41 across five trials.

Boudali et al [116] implemented the least square method for hip and knee joint trajectory prediction. However, this method resulted in larger RMS errors compared to the RNN. Furthermore, it was limited to performing static mappings and could not perform dynamic mappings.

2.7.7 Principal Component Analysis and Best Linear Unbiased Estimation

Complementary limb motion estimation (CLME) is utilised in some wearable robotics, particularly for patients with hemiplegia, in which a healthy limb is used to produce a reference trajectory for a pathological limb based on a mapping function. Vallery et al [117], evaluated CLME using principal component analysis (PCA) or best linear unbiased estimation (BLUE) for the generation of trajectories, comparing it to impedance control (fixed trajectory) and zero-torque control. Evaluation criteria included monitoring the amount of power delivered by the exoskeleton in the different control modes, as well as the distortive impact of this method on muscle activity and the kinematics of gait. Compared to impedance control, CLME produced more natural walking patterns. PCA-CLME also performed worse than BLUE-CLME.

Tanghe et al [97] implemented a probabilistic principal component analysis (PPCA) model for the prediction of gait trajectories. The model is capable of predicting motions over a short time horizon but fails for longer time horizons.

Hassan et al [118] implemented PCA. The authors investigated the efficacy of synergy-based control, also referred to as complementary limb motion estimation (CLME), for patients with hemiparesis. They used the kinematics of the healthy limb and an assistive cane to estimate the reference trajectory of the affected limb. The reference trajectory was the input to a proportional differential (PD) controller, which controls a single-limb HAL exoskeleton. Synergies of the limbs of healthy people were first identified to be able to map the relationship between the healthy and affected limbs of patients with hemiparesis. Synergy-based control was compared to autonomous control.

2.8 Other Parameters

Xu et al [119] conducted a study where they used EEG to decode the intention to perform dorsiflexion for controlling a motorized active foot orthosis. The goal was to promote cortical plasticity for stroke patients, by providing timely control of the dorsi-

flexion of a motorized ankle-foot orthosis. The importance of appropriate stimulation timing correlates with the Hebbian rule that states that 'neurons that fire together, wire together'. The authors used locality-preserving projections (LPP) and a linear discriminant classifier (LDC) to detect movement-related cortical potentials (MRCP). The LPP-MRCP approach performed signal versus noise classification and two consecutive 'signal' classifications indicated the presence of an MRCP. The classifier was tested on 10 healthy subjects, achieving a true prediction rate of 73.0 ± 10.3%. To evaluate whether cortical plasticity was induced, transcranial magnetic stimulation (TMS) was conducted before, right after, and 30 minutes after the 15 min BCI-MAFO session. The study found an 87.2% increase in MEP post-BCI-MAFO, indicating the induction of cortical plasticity.

2.9 Discussion and Conclusion

This systematic literature review presented a comprehensive overview of intelligent algorithms utilised for obtaining gait parameters in the context of controlling wearable lower limb robotics, including exoskeletons and orthoses. Although these technologies share a common objective of assisting individuals in walking and engaging in various locomotion activities, they differ in terms of their control algorithms, functional purpose and sensors employed.

2.9.1 Gait Parameter

Several classification and regression models have been implemented for the identification of gait phases, locomotion modes, torque/moments and joint angle trajectories. The type of parameter that needs to be detected depends on several factors, including:

1. The type of control: Gait phases, for instance, can be used for switching between discrete control modes, while joint trajectory prediction can be used as feedforward to the controller, enhancing stability by compensating for the time delays. Access to future trajectory information or intent can facilitate smoother transitions between control modes [97].

2. The functional purpose of the exoskeleton: Intention inferred from muscle activity can be mapped to a discrete number of control modes. This approach is suitable for neurorehabilitation applications where the intention information serves as a 'trigger' to initiate timely and appropriate movements of the exoskeleton. Other modes of neurorehabilitation or functional purposes may require mapping the intention derived from muscle activity to a continuous signal, such as the desired joint torque [18].

3. The patient's condition and disease: Exoskeletons may be employed for locomotion assistance and/or rehabilitation, serving as compensation for limb amputation or paralysis. They can also promote complete or partial neural recovery in individuals with neurological injuries, including strokes [18].

Considering these factors is crucial when determining the parameters to be detected.

The proportion of the parameters in relation to all of the papers reviewed is as follows: 35.6% gait phase, 31.1% locomotion mode, 24.4% joint angle and trajectory and 8.9% torque and moment (see Figure 2.4).

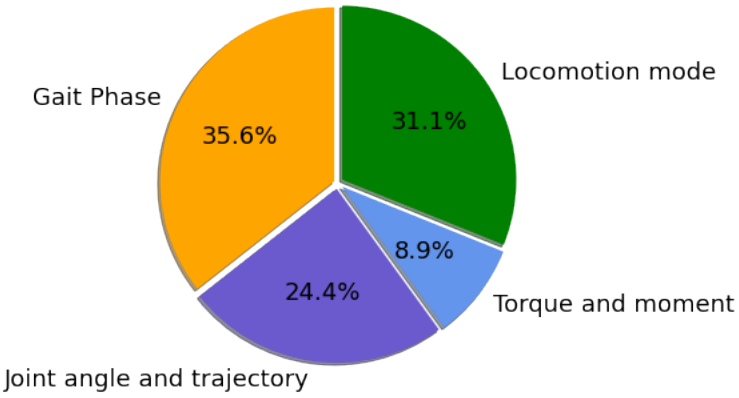

Figure 2.4: Four main gait parameters have been detected or predicted using intelligent algorithms for the control of lower limb robots. The pie chart illustrates the distribution of the gait parameters in the reviewed papers. A large proportion of papers are on gait phases.

Although the maximum number of gait phases is eight, most researchers detected fewer phases in their implementations (see Figure 2.5). Approximately 50% of the papers identified four phases, 30% identified two phases and only one implementation identified all eight phases. Some authors reported that identifying a low number of discrete phases is sufficient for their robotic applications. Another approach presented in some papers

is to determine the percentage of the gait phase instead, which eliminates the need for clear features to identify the start and end of each phase.

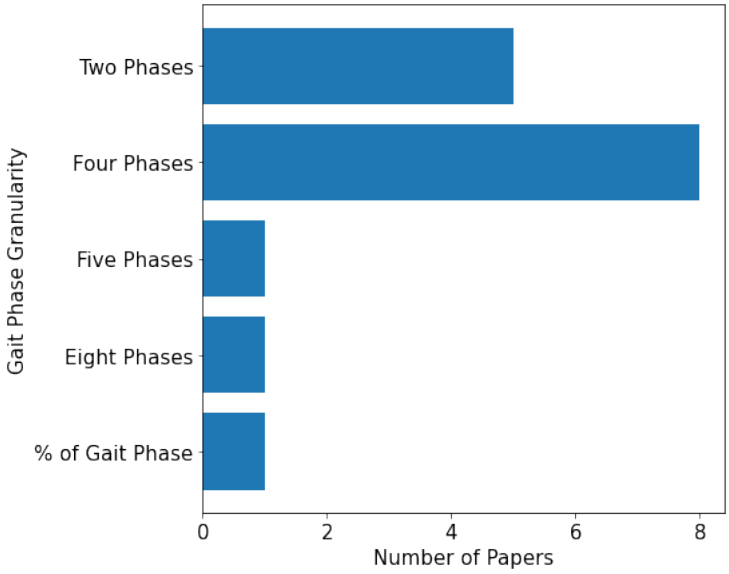

Figure 2.5: Number of phases identified in the reviewed papers

In terms of locomotion mode identification, the number of modes detected varied. Some implementations identified as few as one mode, representing the intention for gait initiation, while others detected up to fifteen modes which include standing, sitting, walking at different speeds, walking up a slope, and other static and dynamic states. Overall, the most commonly identified locomotion modes were level walking, ascending and descending stairs/ramp, and sitting/standing. The number of modes to be detected should be determined based on the device's design, intended purpose, and users.

Several authors focused on joint angles and trajectories. A reference trajectory can be generated and used as input to a controller. Complementary limb motion estimation (CLME) was used in cases where joint trajectories of a healthy limb were utilized to generate trajectories for a pathological limb, such as for patients with hemiparesis. The least commonly detected gait parameter was joint torque/moment, which was only identified in 8.9% of all the reviewed papers. Some papers identified multiple parameters together, mostly combining gait phases with locomotion modes, while the remaining papers focused on detecting one gait parameter only.

2.9.2 Algorithm

Two main types of algorithms have been used for parameter detection/prediction in the reviewed papers. Classification models were used to identify a discrete number of gait phases or locomotion modes, while regression models were used to predict continuous kinematic or kinetic trajectories for estimating joint angles or torques and moments.

In order to compare the performance of various machine learning techniques, some authors implemented multiple algorithms while keeping other influential factors such as sensors, signal processing techniques, and input/testing data constant. A wide range of algorithms was used for gait phase detection, including neural networks, deep neural networks, SVMs, KNN, fuzzy logic etc. The accuracies for gait phase detection ranged from as low as 70% to as high as 98%. Similarly, a wide range of algorithms were used for locomotion mode identification. Most implementations that detected three locomotion modes achieved an accuracy greater than 95%. However, there was a lower diversity of algorithms for torque and moment predictions, which were predominantly predicted with neural networks. Similarly, neural networks were primarily used for the prediction of joint angles and trajectories, followed by PCA.

Deep learning models were implemented in only 15% of the papers reviewed, primarily for gait phases. When deep learning models such as LSTM-DNN, DM-CNN, SSRL, and LSTM were compared to other shallow machine learning implementations, they exhibited higher accuracy and F-score metrics. These promising results encourage further research.

When reporting the performance of the machine learning models, multiple performance metrics have been used, including accuracy, root mean square error (RMSE), F-score, precision and recall rates, as well as delay times. However, not all authors considered the duration of computation and the delay of the prediction. For real-time applications, these factors are crucial to consider when choosing a model or a type of sensor, as even high-accuracy models may not be viable if the delay times or computation durations are significantly large, hindering real-time control of exoskeleton or orthosis.

Different combinations of sensors to measure data were used as input for the models.

Wearable sensors such as IMUs, foot pressure sensors, and EMG electrodes have been utilised in some studies. Non-wearable sensors such as Motion Capture Systems, and ground force plates (measure ground reaction force) have also been used. Although smartphone sensors have been used in gait analysis [120], they were not deployed in the papers reviewed in this study. Each sensor type has its advantages and disadvantages. IMUs which measure acceleration signals suffer from drift, but this issue has been addressed by calibrating with every step. A method to address this issue has been proposed by Qui et al [121].

There were single-modality and multi-modality approaches. Single-modality approaches rely on data from a single sensor such as an IMU or EMG, while multi-modality approaches combine data from multiple sensors. The optimal location for sensor placement has been frequently studied [122]. Preferences for specific sensing modalities have emerged based on the predicted parameter. IMU sensors have been commonly used for gait phase detection, whereas EEG has been commonly used for locomotion modes and not for any other parameter. EMG has predominantly been used for torque/moment and joint angle trajectory predictions.

It's important to differentiate between sensors that collect measurements as input for algorithms and those that collect measurements to segment data or produce ground truth labels for supervised learning algorithms. Foot pressure sensors have been commonly used for the segmentation or labelling of gait phases. Measurements from motion capture systems have served both as inputs to ML algorithms or as references for evaluating predictions of ML algorithms, as motion capture systems are considered the 'gold standard' for certain measurements.

There have been varied choices regarding the use of accelerometers, gyroscopes, or both as well as the number of axes considered. Also, multiple sensors can be used to obtain the same measurement. For example, joint angles could be measured with a goniometer or motion capture system, but they can also be derived using signals from IMU using quaternion calculations. The choice of sensors can impact performance when comparing wearable and non-wearable sensors. Thus, algorithms trained with data from motion capture systems may demonstrate lower performance when data from wearable sensors

are used in the actual implementation.

The type of sensor used has an impact on the detection of locomotion mode. Implementations involving EEG or EMG sensors have demonstrated lower prediction accuracies, ranging from 68.4% to 92.4%, while implementations utilising other sensors such as IMUs had accuracies mostly above 90%. For torque and moment prediction, EMG and joint angle sensors were predominantly used.

2.9.3 Training Dataset

In the identified studies, the number of participants in the testing of the proposed models is generally small, with some cases having as little as one participant. This can pose an issue, particularly for generalised models that aim to capture inter-subject variability. The highest number of participants in a study reached around thirty, while others ranged around ten participants. Studies have shown that algorithms trained on one group of participants and tested on data from other participants perform worse than personalised models that learn the gait patterns from a single person and are tested on unseen data from that person. Ideally, achieving higher performance with a generalised model is desirable, yet if inter-subject variability remains high, even among people with similar anthropometrics and gait conditions, individual models may be necessary.

The environmental conditions in which the training sets were recorded also impact the real-time performance of the algorithms. Most of the data collected comes from experiments in labs, usually under controlled environments. Some studies set walking speeds (using treadmills), while others involved self-selected speeds. These conditions may perturb natural gait patterns, which typically experience more stochasticity and noise in real environments. Therefore the performance of some of the models presents an "upper-bound on the performance" [123], and some deterioration in performance should be expected.

2.9.4 Future Directions

The development of robotic devices and their control models has been progressing rapidly. This chapter presents a range of state-of-the-art models that have been used in controlling exoskeletons, highlighting the different parameters that researchers have been investigating. Both classification and regression methods have been successfully integrated with exoskeleton controllers, as demonstrated by the papers applying these algorithms. As previously discussed, the choice of methods depends on the type of controller, the functional purpose of the exoskeleton, and the disease and condition of the patient. However, further research is needed to compare and contrast these methods and investigate their integration with controllers.

One of the important issues that need to be addressed is the lack of pathological gait datasets. This is a known limitation across the majority of the papers, as the models presented require training and testing on pathological gait patterns. In the presence of pathological gait data, transfer learning can be used to enhance the model's adaptability to different gait patterns. Knowledge gained in identifying healthy gait parameters can be transferred to pathological gait, particularly when datasets for users with pathologies are limited [124, 125]. It is not only pathological gait data that is needed, but also representative data for accurate predictions. The data sets should include gait patterns obtained while users are wearing an exoskeleton, in zero-torque mode, or in assistive modes as these conditions alter gait patterns. They should also include users walking at a range of speeds.

Furthermore, the classification models used for gait phase or locomotion mode detection primarily focus on identifying activities. It would be beneficial to develop algorithms capable of assessing the activity [126] since this would enable evaluating the current state of the user and monitoring how their gait progresses with the use of the robotic device. The development of explainable AI is also important, allowing us to gain insights into the factors that influence predictions and enhance our understanding of gait analyses. For instance, explainable AI is particularly significant in decoding intention from brain signals, as this will increase our understanding of the brain, and the role of its regions.

Human-robot interaction requires awareness not only of the state of the user but also of the environment. Enhancing environmental awareness is crucial, such as developing models capable of detecting the type of terrain the user is walking on and adapting the assistance accordingly. This would enhance the performance of robotic devices in real-world scenarios.

Given the limited amount of research on the use of AI for gait trajectory predictions, accounting for only 24.4% of the reviewed studies compared to the other gait parameters, and recognising the value of incorporating trajectory information in exoskeleton control strategies, the subsequent chapters will primarily focus on predicting gait trajectories using AI methods. Each chapter will introduce a study regarding this topic and provide further relevant background information.

Chapter 3: Datasets & Ethical Considerations

3.1 Overview

This chapter introduces the three datasets used to develop (train and evaluate) the trajectory forecasting models presented in Chapters (4 to 6). The Canterbury Christ Church University & Chailey Clinic dataset was collected by partner institutions which we collaborated with for this research, while the Gillette Children Specialty Healthcare dataset and the EPIC - Georgia Tech University dataset are publicly available online. The chapter also discusses ethical considerations regarding the data used.

3.2 Datasets

3.2.1 Gillette Children Specialty Healthcare Dataset

The Gillette Children Speciality Healthcare dataset was used for the development of the models presented in Chapter 4. The deep learning models implemented were trained and evaluated on data from children with neurological disorders. The data is available online and was collected over the years 1994 and 2017 [127]. It has been previously used for the development of a deep learning model aimed at automatically detecting gait events, specifically foot contact and foot off events [25]. The dataset comprises recordings of children between the ages of 4 and 19. The majority of them had cerebral

Table 3.1: Demographics of the Children in Gillette Children Specialty Healthcare Dataset [25]

	Training Set	Test Set
Age (years)	11.4 (std = 6.2)	11.0 (std = 4.5)
Weight (kg)	35.7 (std = 17.7)	35.9 (std = 16.7)
Height (cm)	135.7 (std = 21.6)	135.6 (std = 21.4)
Leg length (cm)	70.3 (std = 14.0)	70.6 (std = 12.8)
Walking speed (m/s)	0.84 (std = 0.28)	0.85 (std = 0.29)

palsy (73%), while the remaining had a combination of neurological, developmental, orthopaedic, and genetic disorders (27%). A motion capture system (VICON) was used for data collection, with a sampling frequency of 120 Hz. The children were recorded while walking for a distance of 15 meters and the data consist of a 99-dimensional vector containing kinematics and marker positions.

For this study, the data were divided into a training and testing set. The statistical distribution of the features in the training set is as follows: age (11.4 $\pm$ 6.2 years), weight (35.7 $\pm$ 17.7 kg), height (135.7 $\pm$ 21.6 cm), leg length (70.3 $\pm$ 14.0 cm), and walking speed (0.84 $\pm$ 0.28 m/s). The statistical distribution of the children in the testing set is as follows: age (11.0 $\pm$ 4.5 years), weight (35.9 $\pm$ 16.7 kg), height (135.6 $\pm$ 21.4 cm), leg length (70.6 $\pm$ 12.8 cm), and walking speed (0.85 $\pm$ 0.29 m/s) [25]. In this study, only kinematics were used for trajectory forecasting; these include angles of the hip, knee, and ankle in the yaw, pitch, and roll dimensions. The kinematic data represent Euler angles, which are calculated using the plug-in-gait mechanical model [25].

3.2.2 Canterbury Christ Church University & Chailey Clinic Dataset

A dataset containing gait recordings from both typically developing children and children with Cerebral Palsy was used to develop and evaluate the models in Chapter 5. The dataset includes flexion-extension angles of the hip, knee and ankle measured simultaneously in the sagittal plane, for the right and left legs. The data was collected and provided by Canterbury Christ Church University, and Chailey Clinical Services, two partners in the MOTION project which we collaborated with for this research.

Table 3.2: Distribution of the Children in the Canterbury Christ Church University & Chailey Clinic Dataset

	Number of Children	Age (years)	Height (m)	Mass (kg)
Typically Developing	10	8.10 ± 2.56	133.39 ± 16.25	29.85 ± 9.76
Cerebral Palsy	11	9.45 ± 1.37	135.25 ± 8.29	33.44 ± 10.07

The gait of typically developing (TD) children and children with Cerebral Palsy (CP) was recorded while they walked at self-selected speeds. Several trials were conducted for each child, with a requirement to walk a distance of 8 meters per trial. Data were collected using the ISEN inertial motion capture system (STT Systems, Spain) which uses inertial measurement units (IMUs) to capture the gait (refer to Figure 3.1 for locations of IMU placement). The raw inertial measurements collected by IMUs were exported and processed by the accompanying ISEN software, which derived the flexion-extension angles for the hip, knee, and ankle, for the left and right legs. The data were collected at a sampling frequency of 100Hz.

The study included a total of 10 typically developing (TD) children and 11 children with Cerebral Palsy (CP) (see Table 3.2). TD children were between 4 and 13 years old, while children with CP were between 8 and 12 years old with a Gross Motor Function Classification Scale (GMFCS) between I-II. The anthropometrics and demographic details of the participants are included in Figure 3.2 and Figure 3.3 respectively.

3.2.3 EPIC - Georgia Tech University Dataset

The models in Chapter 6 were developed using an online gait dataset by Camargo et al. [128]. The dataset consists of gait data collected from 22 able-bodied individuals (age 21 ± 3.4 years, height 1.70 ± 0.07 m, mass 68.3 ± 10.83 kg). Gait patterns were recorded while walking on a treadmill at 28 different speeds (0.5 m/s to 1.85 m/s in 0.05 m/s increments). The dataset contains joint kinematic values including the hip, knee, and ankle angles of the left and right legs in the sagittal plane, which were used in this study. The joint angles were derived using OpenSim's inverse kinematics

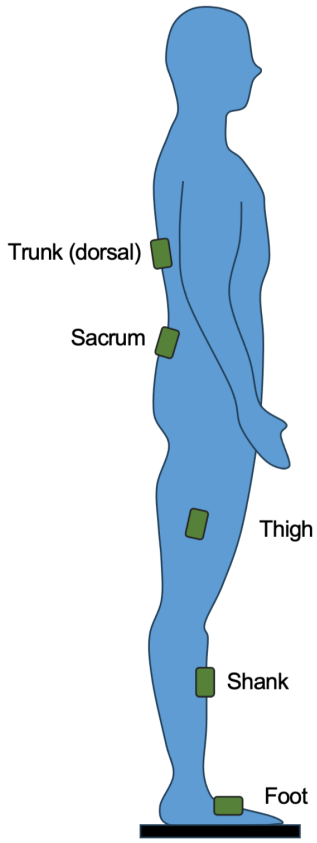

Figure 3.1: Locations of IMU sensor placement for the Canterbury Christ Church University & Chailey Clinic Data Collection. IMUs were placed on the feet, shank, thigh, sacrum, and trunk (dorsal) according to the ISEN inertial motion capture system (STT Systems) guidelines.

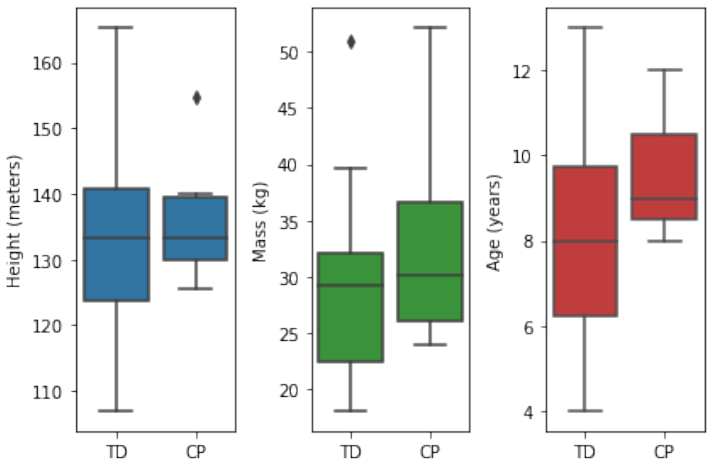

Figure 3.2: Box plot of the anthropometrics of Typically Developing (TD) children and children with Cerebral Palsy (CP) in the Canterbury Christ Church University & Chailey Clinic Dataset.

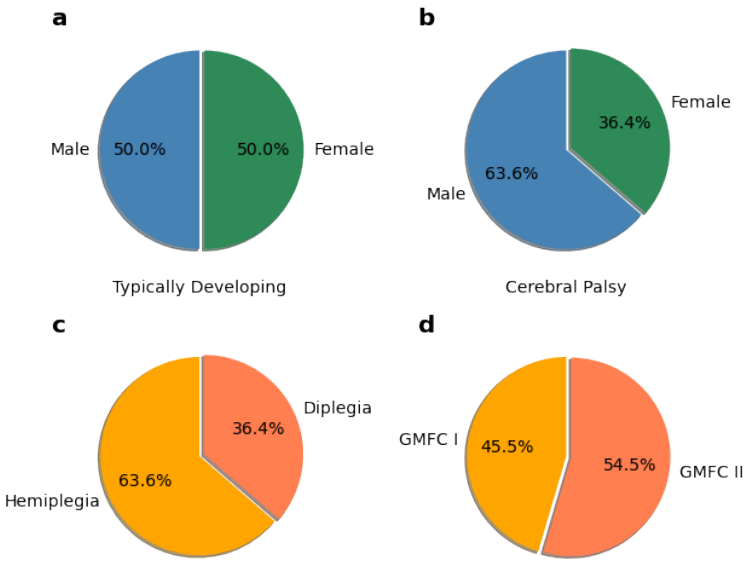

Figure 3.3: Demographics of Typically Developing children and children with Cerebral Palsy in the Canterbury Christ Church University & Chailey Clinic Dataset (in percentages). (a) and (b) report the gender of typically developing children and children with Cerebral Palsy respectively, (c) is the type of CP, and (d) is the Gross Motor Function Classification of children with CP.

tool based on motion capture data collected at a 200 Hz sampling frequency. For a visual representation of the demographic distribution of the dataset, refer to Figure 3.4. Additional details about the data processing procedures can be found in [128].

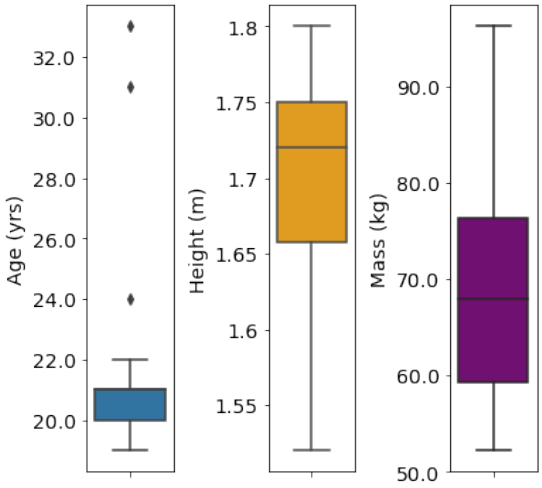

Figure 3.4: Box-plots showing demographics of the individuals of the training and testing sets in the EPIC - Georgia Tech University Dataset, which includes their age in years, height in meters, and mass in kilograms

3.3 Ethical Considerations

The Gillette Children Speciality Healthcare dataset (used in Chapter 4, available at [127]) and the EPIC - Georgia Tech University dataset (used in Chapter 6, available at [128]) are publically available online, with the participants anonymised to preserve their privacy. Regarding the dataset collected by our research collaborators, Canterbury Christ Church University & Chailey Clinic (used in Chapter 5), local ethical approval and NHS approval were granted for these studies (IRAS project ID 288842, reference number 21/PR/0927, CCCU ethics: ETH1920-0278).

Chapter 4: Forecasting Gait Trajectories of Children with Neurological Disorders

4.1 Overview

This chapter is focused on the implementation of deep learning models for gait trajectory prediction of children with neurological disorders. The dataset used to train the models contains gait recordings of children with neurological disorders, 73% of which have Cerebral Palsy. The performances of two deep learning models, a long-short-term memory network and a convolutional neural network, are compared to two non-intelligent models. Furthermore, this chapter presents the results of investigating the influence of varying the length of the input and output (prediction) window on the predictive performance of the models.

4.2 Introduction

Powered exoskeletons require a control strategy to guide the exoskeleton's interaction with the user. This control strategy coordinates the movement of the exoskeleton with the user's body providing full support or synchronises the exoskeleton's movement with the user's body providing partial support [17]. As discussed in chapter 2, the control strategy typically follows a three-level hierarchy: high, mid, and low levels [17]. At the high level, the control strategy is responsible for detecting user intention and predicting the desired state of the exoskeleton [17, 19, 52]. This includes estimating desired torque

from EMG signals [106], classifying locomotion modes (e.g., standing up, sitting down, walking up/down a staircase, etc.) [98] and their transitions, as well as environment classification (predicting the user's interaction with the surrounding environment [129]). The mid-level control is responsible for selecting and switching between the continuous states of the exoskeleton [17, 52]. Baud et al. [17], in their review on control strategies for lower limb exoskeletons, divided this level into detection/synchronisation and action sub-levels. Detection/synchronisation involves identifying the state of the user, such as the phase of gait, while action computes the appropriate output for the identified state. The low-level control directly controls the actuators of the exoskeleton and implements the control strategy by supplying torque to the joints. It tracks the reference input and ensures stability [51–53].

The timing and magnitude of supportive torque provided by an exoskeleton depend on the control strategy implemented, which is influenced by the specific application of an exoskeleton. In rehabilitation applications, one common control strategy is *trajectory tracking*, where the exoskeleton follows a predefined trajectory, that can be based on the trajectory of a healthy user [130]. Another strategy used in rehabilitative exoskeletons is *assis-as-needed*, where the level of support provided by the exoskeletons is variable and can change throughout the course of rehabilitation. An example of an assist-as-needed strategy is impedance control, where the assistance provided depends on the effort exerted by the patient [130]. In locomotion assistance applications, trajectory tracking is also commonly used. The predefined trajectory could be from a healthy user, or in the case of hemiparetic patients, the predefined trajectory for the pathological limb can be from the healthy limb (also known as complementary limb motion estimation [117]). As for strength augmentation applications, hybrid and force control strategies are commonly employed [130].

Integrating gait trajectory prediction into the control strategy of exoskeletons is an area of research [97]. Several studies investigated the use of deep learning techniques for gait trajectory prediction. Liu et al. [131] developed a deep spatio-temporal model using Long Short-Term Memory (LSTM) units to predict two time-steps in the future, and averaged the predictions to smooth fluctuations. Zaroug et al. [132] implemented an auto-encoder LSTM for predicting linear acceleration and angular velocity trajectories.

They experimented with varying lengths of input time steps, between five and 40 steps, to predict five or 10 steps in the future (equivalent to 30 ms or 60 ms). An LSTM with a weighted discount loss function was proposed by Su et al. [133] for predicting angular velocities of the thigh, shank, and foot segments. They used 10 or 30 time-steps as input to predict five or 10 steps in the future, corresponding to 100 ms and 200 ms, respectively. Hernandez et al. [134] used a hybrid Convolutional Neural Network (CNN) and LSTM neural network, DeepConvLSTM, to forecast kinematic trajectories with an average MAE of 3.6°. Jia et al. [135] implemented a deep neural network with LSTM units and a feature fusion layer to combine kinematic and physiological data (i.e., joint angles and EMG) for trajectory prediction. Zarough et al. [136] also compared vanilla, stacked, bidirectional, and autoencoder LSTMs while Zhu et al. [137] used attention-based CNN-LSTM, predicting trajectories 60 ms in the future.

It has been observed that the values of kinematic parameters within a gait cycle exhibit greater variation between different individuals (inter-subject) compared to the values of kinematic parameters within the same individual (intra-subject) [138]. Studies have shown that intra-subject trajectory prediction models, which are tested on data from the same individuals used for training, tend to be more accurate in their predictions than inter-subject models, which are tested on data from individuals who were not included in the training process [133]. However, the variability of pathological gait, such as in children with spastic cerebral palsy, is even higher compared to healthy gait. Children with spastic cerebral palsy were found to have higher within-day and between-day variability compared to healthy children, possibly attributed to spasticity that limits their joint range of motion [139, 140].

Existing studies have primarily focused on models trained on healthy gait trajectories only, the ability of deep learning models to accurately forecast pathological trajectories with greater heterogeneity and variability has yet to be evaluated. The main contribution of this study is to investigate, for the first time, the performance of deep learning networks, specifically the Long Short-Term Memory (LSTM) neural network and Convolutional Neural Network (CNN), in forecasting pathological gait trajectories of children with neurological disorders. A comparison between the two networks is conducted. Furthermore, the influence of the length of the input and output windows

on prediction accuracy is investigated, while providing technical recommendations for improving the prediction process.

4.3 Materials and Methods

4.3.1 Data

The deep learning models implemented in this study were trained and evaluated on the Gillette Children Specialty Healthcare Dataset. Details on this dataset can be found in Chapter 3, section 3.2.1.

4.3.2 Data Processing

The available data included Euler angles of the hip, knee, and ankle in yaw, pitch, and roll dimensions for both legs, however, data from only one leg was utilised for training the models. The pre-processing of data involved trimming the leading and the trailing zeros, and removing trials with spurious data (which were assumed to be Euler angles greater or less than a 90° cut-off threshold).

To train deep learning models, fixed-length input and target trajectory sequences were generated using the sliding window method, illustrated in Figure 4.1. This method involves creating a shorter sequence x_{in} with k time-steps, where k is the size of the input, i.e., the number of time-steps utilised by the model to make predictions, $x_{in} = \{x_1, x_2, \ldots, x_k\}$. A corresponding shorter output sequence y_{out} with z time-steps is created, where z is the size of the output, i.e., the number of time-steps that will be predicted by the model, $y_{out} = \{x_{k+1}, x_{k+2}, \ldots, x_{k+z}\}$. Each input window has a corresponding output window (target label to train the models), and together they form one training sample. The stride, which denotes the distance between the beginning of one sample and the beginning of the next sample, was set to 5 time-steps. Ten samples are generated from each trial.

The models were trained using various sizes of input and output windows. For the

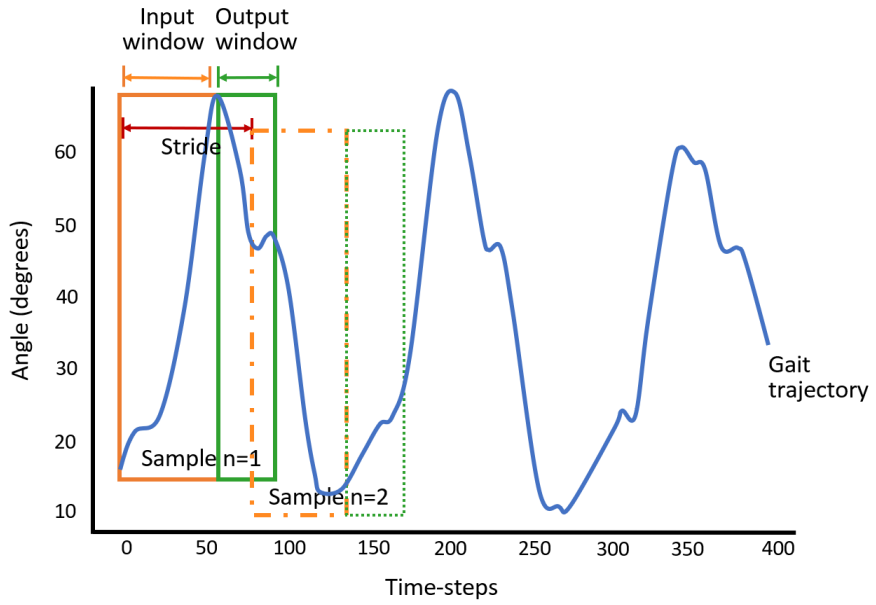

Figure 4.1: Illustration of the sliding window method. Continuous gait trajectories are used to generate input and output windows for training the model using the sliding window method. Each pair of input and output windows forms one sample. Several samples can be generated from one continuous gait trajectory by sliding the window for a specified distance, also known as stride length.

LSTM model, the input window sizes were 50, 100, 200, 400, 600, 800, and 1000 ms, corresponding to 6, 12, 24, 48, 72, 96, and 120 time-steps based on the data's sampling frequency of 120Hz. The output window sizes for the LSTM were 8.33, 25, 50, 100, and 200 ms which correspond to 1, 3, 6, 12, and 24 time-steps. This resulted in a total of 35 combinations of input and output window sizes. As for the CNN, the focus was on using 6 and 120 input time-steps (the smallest and largest input sizes we used with the LSTM) to predict 1, 3, 6, 12, and 24 output time-steps. These time ranges were chosen based on values proposed in the literature by researchers that forecasted healthy gait. Zaroug et al. experimented with a wide range of input windows from 5 to 40 steps to forecast 5 or 10 steps in the future, which correspond to 30 and 60 ms respectively [132]. Output windows ranging from 50 to 60 ms were common too [132, 135, 137], and a few researchers have predicted output windows of 100 ms or larger [133, 136].

For n training samples, f features (hip, knee, and ankle angles in the yaw, pitch, and roll directions), l_{in} input window size and l_{out} output window size, an input matrix X and target output matrix Y are created, where $X \in \mathbb{R}^{n \times l_{in} \times f}$ and $Y \in \mathbb{R}^{n \times l_{out} \times f}$. The matrices are then normalised such that $X \in [0, 1]$ and $Y \in [0, 1]$. The aim is to build a model $g()$ that maps X to $\hat{Y}$, i.e., $\hat{Y} = g(X)$, where $\hat{Y}$ is a close approximation to true

value Y.

4.3.3 Long Short-Term Memory (LSTM) Architecture

The Long Short-Term Memory (LSTM) neural network is a popular choice for time-series applications including forecasting future trajectories. It is a type of gated Recurrent Neural Network (RNN), which addresses the issue of vanishing and exploding gradients during training and is capable of learning long-term dependencies in the data [141]. An LSTM cell contains three gates: an input gate, an output gate, and a forget gate. The equations of these gates, as explained in Goodfellow et al. [141], are reported below.

The forget gate unit $f_i^{(t)}$ equation for time-step t, cell i, input vector $x^{(t)}$, hidden layer vector $h^{(t)}$, biases b^f, recurrent weights W^f, and input weights U^f is:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right). \tag{4.1}$$

The sigmoid function, represented by σ, bounds the value between 0 and 1.

The internal state of an LSTM cell, $s_i^{(t)}$, is updated depending on the value of the forget gate unit $f_i^{(t)}$, and its equation for biases b, input weights U, and recurrent weights W is:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right). \tag{4.2}$$

The external input gate unit, $g_i^{(t)}$, is calculated as:

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right). \tag{4.3}$$

The output of the LSTM cell $h_i^{(t)}$ is calculated as:

$$h_i^{(t)} = tanh\left(s_i^{(t)}\right) q_i^{(t)}. \tag{4.4}$$

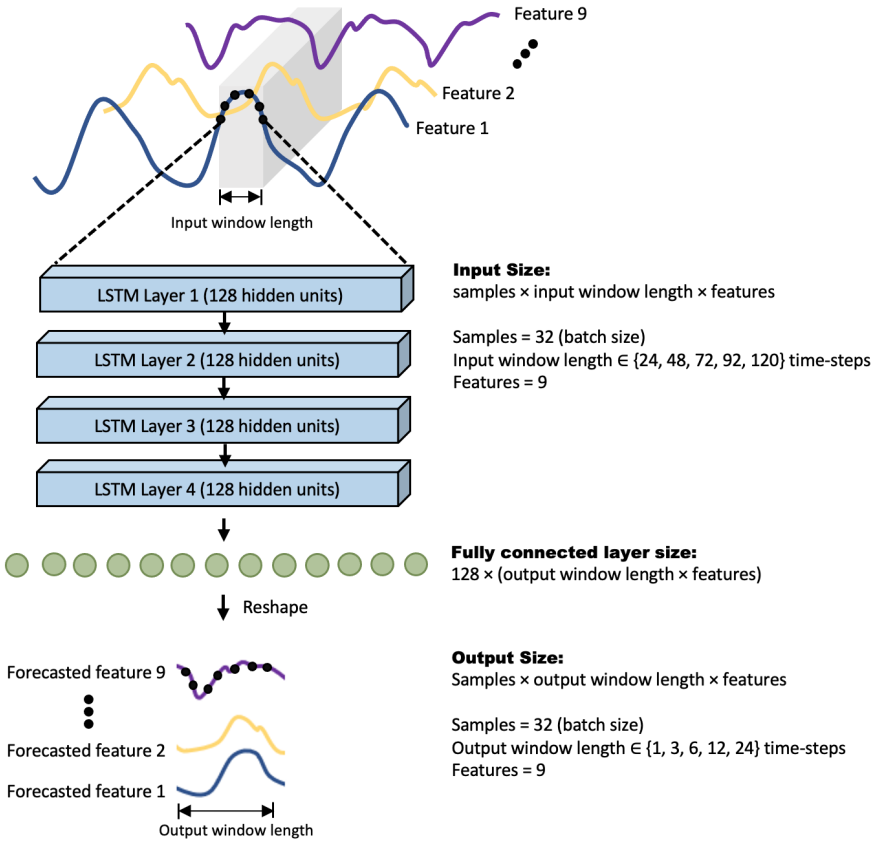

Figure 4.2: Architecture of the long-short-term memory (LSTM) network used in this study. The model consists of four LSTM layers with 128 hidden units each, followed by a fully connected layer. The input and output features of the LSTM include Euler angles of the hip, knee, and ankle in the yaw, pitch, and roll dimensions.

For biases b^o, input weights U^o, and recurrent weights W^o, the value of the output gate unit is:

$$q_i^{(t)} = \sigma\left(b_i^o, + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right). \tag{4.5}$$

The LSTM network implemented in this study contained 4 layers of LSTM units, with 128 units per layer. The hidden state of the final layer is used as input to a fully connected layer which is then reshaped to obtain the desired output. The overall architecture is depicted in Figure 4.2.

4.3.4 Convolutional Neural Network (CNN) Architecture

In this study, a Convolutional Neural Network (CNN) is used to map input trajectories X to forecasted predictions $\hat{Y}$. The CNN utilizes the convolution operation between

a sequence and a kernel, with the weights of the kernel adjusted during the learning process. Equation (4.6) is employed to calculate the output of the convolution operation S, which has been adapted from the format described in Goodfellow et al. [141] to accommodate a 1D time-series I and kernel K.

$$S(i) = (I * K)(i) = \sum_{m} I(m)K(i - m). \tag{4.6}$$

The CNN architecture consists of several 1D convolutions and pooling layers, followed by a dense fully connected layer. The number of kernels in each convolution layer, as well as the size of the pooling layers, is illustrated in Figure 4.3. Dilated convolutions were experimented with, but they did not improve performance, so they were not included in the final architecture.

4.3.5 Baseline Methods

The performance of the deep learning models was benchmarked against a simpler machine learning model, a Fully Connected Network (FCN), and two non-intelligent models, referred to as Naïve Method 1 and Naïve Method 2.

The FCN implemented in this Chapter, and in Chapters 5 and 6 are dense feedforward neural networks, sometimes referred to as multilayer perceptions (MLP) in literature. They consist of several linear layers with an activation function in between (ReLU) and are trained using gradient descent and the Adam optimiser. The only difference between the FCN implemented in Chapters 4, 5, and 6 is the number of units in each linear layer, as well as the overall number of layers. However, gradient descent and Adam optimiser have been used for training the FCNs in all those chapters. In this study, the FCN contains five hidden layers with 200 nodes per layer.

The first naïve method predicts the same value for all output time-steps, which is equal to the final time-step in the input window. The second naïve method predicts the mean value of the input as the predicted value for all output time-steps.

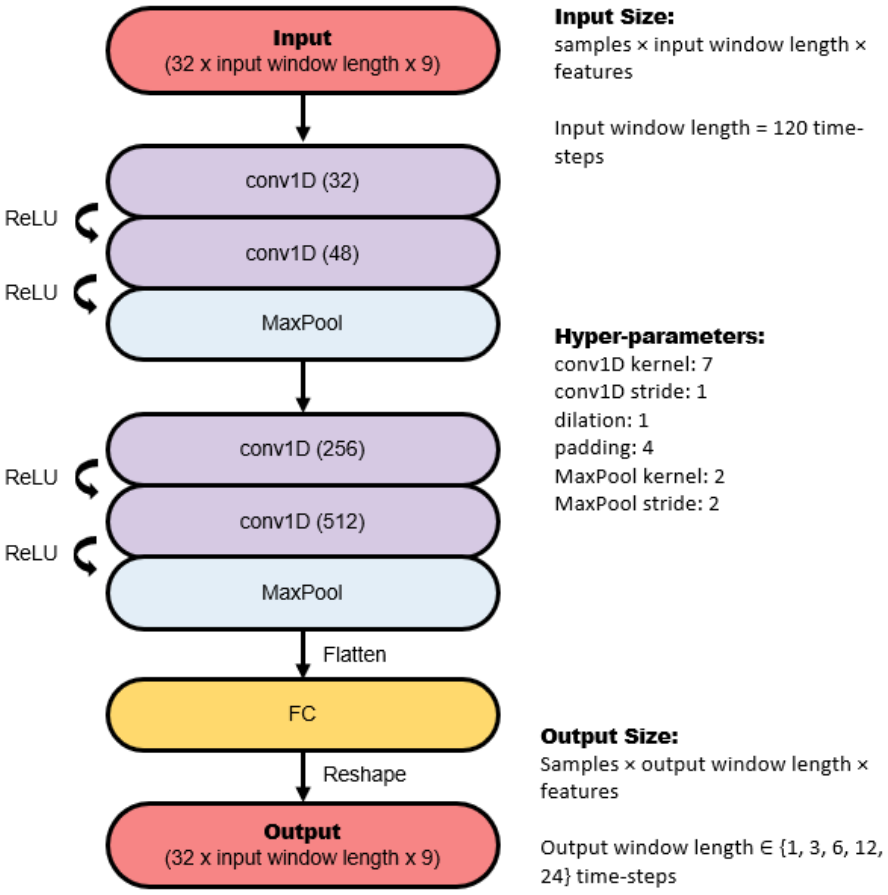

Figure 4.3: Architecture of the convolutional neural network (CNN) used in this study. The model consists of four 1-dimensional convolutional layers, two pooling layers, and a fully connected layer. The input and output features of the CNN include Euler angles of the hip, knee, and ankle in the yaw, pitch, and roll dimensions

4.3.6 Details of Network Implementation

The objective of the deep learning models in this study is to find a mapping between the input trajectories X and forecasted trajectories $\hat{Y}$, such that the error between predicted trajectories $\hat{Y}$ and true trajectories Y is minimised. The loss function that has been used to optimise the deep learning models is the Mean Squared Error (MSE).

The number of trials used from the dataset was 16,782 and each trial was used to extract 10 samples using the sliding window method, described in section 4.3.2. This resulted in a total of 167,820 samples. The samples were split into training (70%), validation (20%), and testing (10%) sets. Both the CNN and LSTM were trained in mini-batches, with a batch size of 32 samples. The Adam optimiser was used for learning.

To determine the hyper-parameters and architecture for the CNN, LSTM, and FCN models (described in section 4.3.3 and section 4.3.5), a hyper-parameter search was performed. The search involved defining a search space for the parameters and selecting the parameters that optimize the performance of the network by minimizing the validation loss. The parameters in the search included learning rate, number of layers of LSTM, number of hidden units, and size of kernels. The tree-structured Parzen estimator algorithm, a type of Bayesian hyper-parameter sampler, was used for this purpose [142]. The search space for the hyper-parameters and the corresponding selected values can be found in Table 4.1. The hyperparameters were optimised for predictions with an input window size of 72 time-steps and an output window size of 12 time-steps. The number of epochs, which was also included in the hyper-parameter search, was fine-tuned manually afterward. The numbers of epochs for training the LSTM, CNN, and FCN were 60, 150, and 180, respectively. Once the optimal architecture and parameters were selected, the training and validation sets were combined to train these networks. The total numbers of trainable parameters for the optimised LSTM, CNN, and FCN models were 495,320, 4,666,888, and 380,216, respectively. The final performance of the networks was evaluated on the testing set.

The framework described in this study has been implemented using Python, utilising the following libraries: Pytorch, Numpy, Matplotlib, SciPy, and Scikit-learn. Optuna

Table 4.1: Hyper-parameter optimisation for the LSTM network, CNN, and FCN.

	Hyper-Parameter	**Search Space**	**Selected Value**
LSTM	learning rate	[0.1, 0.01, 0.001, 0.0001, 0.00001]	0.001
	number of LSTM layers	[1, 2, 3, 4]	4
	number of LSTM hidden units	[16, 32, 64, 100, 128]	128
CNN	learning rate	[0.1, 0.01, 0.001, 0.0001, 0.00001]	0.0001
	conv1D layer 1 channels	[16, 32, 48]	32
	conv1D layer 2 channels	[32, 48, 64]	48
	conv1D layer 3 channels	[64, 128, 256]	256
	conv1D layer 4 channels	[128, 256, 512]	512
	kernel size for layers 1, 2	[1, 2, 3, 4, 5, 6, 7]	7
	kernel size for layers 3, 4	[1, 2, 3, 4, 5, 6, 7]	7
	padding	[0, 1, 2, 3, 4, 5]	4
	conv1D stride	[1, 2, 3, 4, 5]	1
	dilation	[1, 2, 4]	1
FCN	learning rate	[0.1, 0.01, 0.001, 0.0001, 0.00001]	0.001
	hidden layers	[3, 4, 5, 6, 8, 10, 12]	5
	nodes per layer	[10, 20, 40, 60, 100, 140, 160, 200]	200

was used for the hyper-parameter search [142]. An Nvidia Geforce RTX 2070 GPU was used for computation.

4.3.7 Evaluation Metrics

Several metrics have been used to evaluate the performance of the models. To measure how close the predicted trajectories $\hat{Y}$ are to the observed trajectories Y the mean square error MSE, mean absolute error MAE, and Pearson correlation coefficient P were calculated after the de-normalisation (i.e., re-scaling to the original ranges) of the predicted trajectories. Given that n is the number of testing samples, f is the number of features, and l_{out} is the output window size, the equations are as follows:

Mean absolute error (MAE):

$$MAE = \frac{1}{n.f.l_{out}} \sum_{i=1}^{n} \sum_{j=1}^{f} \sum_{k=1}^{l_{out}} |y_{i,j,k} - \hat{y}_{i,j,k}| \tag{4.7}$$

Mean absolute error (MAE) standard deviation:

$$\sigma_{MAE} = \sqrt{\frac{1}{n.f.l_{out}} \sum_{i=1}^{n} \sum_{j=1}^{f} \sum_{k=1}^{l_{out}} (|y_{i,j,k} - \hat{y}_{i,j,k}| - MAE)^2} \tag{4.8}$$

Mean squared error (MSE):

$$MSE = \frac{1}{n.f.l_{out}} \sum_{i=1}^{n} \sum_{j=1}^{f} \sum_{k=1}^{l_{out}} (y_{i,j,k} - \hat{y}_{i,j,k})^2 \tag{4.9}$$

Mean squared error (MSE) standard deviation:

$$\sigma_{MSE} = \sqrt{\frac{1}{n.f.l_{out}} \sum_{i=1}^{n} \sum_{j=1}^{f} \sum_{k=1}^{l_{out}} ((y_{i,j,k} - \hat{y}_{i,j,k})^2 - MSE)^2} \tag{4.10}$$

Pearson correlation coefficient:

$$P = \frac{1}{f} \sum_{j=1}^{f} \frac{cov(y_j, \hat{y}_j)}{std(y_j) \times std(\hat{y}_j)}. \tag{4.11}$$

These metrics were used to evaluate and compare the performance of the networks, with results presented in Section 4.4.

4.4 Results

4.4.1 LSTM Network Performance for Varying Input and Output Window Sizes

The LSTM model was trained using 35 combinations of input and output window sizes. The input window sizes were 6, 12, 24, 48, 72, 96, and 120 time-steps, which correspond to 50, 100, 200, 400, 600, 800, and 1000 ms, respectively. The output window sizes were 1, 3, 6, 12, and 24 time-steps, and correspond to 8.33, 25, 50, 100, and 200 ms. The performance of the models measured by MAE, MSE, and Pearson correlation coefficient is reported in Table 4.2.

The results in Table 4.2 show that as the size of the output window decreases, the prediction errors also decrease. LSTMs predicting one output time-step had the lowest mean errors, while LSTMs predicting 24 time-steps had the highest mean errors, which was expected. On the other hand, the size of the input window did not have a significant influence on mean losses when predicting short output windows, specifically six output time-steps or below (see Figure 4.4a-c). However, the size of the input window had an influence when predicting larger output windows of 12 and 24 time-steps. Larger input sizes resulted in lower mean errors (see Figure 4.4d,e). Specifically, for 12 and 24 time-step output windows, using a 120 time-step input window size, which is the largest input size tested, led to the lowest mean absolute errors (see Figure 4.4a-c).

Table 4.2: Performance of LSTM in forecasting gait trajectories for varying input and output window sizes.

Input Window Size (ms)	Input Time-Steps	Output Window Size (ms)	Output Time-Steps	MSE (Degrees)	MSE std (Degrees)	MAE (Degrees)	MAE std (Degrees)	Mean Pearson Correlation Coefficient
50	6	8.33	1	0.034	0.065	0.143	0.115	1.000
100	12	8.33	1	0.077	0.130	0.214	0.177	1.000
200	24	8.33	1	0.027	0.055	0.126	0.105	1.000
400	**48**	**8.33**	**1**	**0.019**	**0.161**	**0.095**	**0.099**	**1.000**
600	72	8.33	1	0.030	0.266	0.126	0.119	1.000
800	96	8.33	1	0.020	0.125	0.107	0.092	1.000
1000	120	8.33	1	0.022	0.318	0.109	0.098	1.000
50	6	25	3	0.079	0.526	0.175	0.220	1.000
100	12	25	3	0.077	0.474	0.187	0.206	1.000
200	24	25	3	0.079	0.793	0.176	0.218	1.000
400	**48**	**25**	**3**	**0.080**	**3.068**	**0.169**	**0.227**	**1.000**
600	72	25	3	0.092	2.597	0.173	0.250	1.000
800	96	25	3	0.104	1.279	0.200	0.252	1.000
1000	120	25	3	0.117	2.028	0.223	0.261	1.000
50	6	50	6	0.614	4.115	0.461	0.633	0.998
100	12	50	6	0.422	3.956	0.365	0.537	0.998
200	24	50	6	0.416	4.416	0.350	0.541	0.998
400	48	50	6	0.381	2.773	0.356	0.505	0.998
600	72	50	6	0.426	7.653	0.352	0.550	0.998
800	**96**	**50**	**6**	**0.363**	**3.377**	**0.332**	**0.502**	**0.998**
1000	120	50	6	0.405	5.414	0.359	0.526	0.998
50	6	100	12	3.548	15.749	1.104	1.526	0.984
100	12	100	12	3.310	15.545	1.058	1.480	0.985
200	24	100	12	3.200	14.790	1.008	1.478	0.985
400	48	100	12	3.279	30.567	1.007	1.505	0.985
600	72	100	12	2.723	14.993	0.927	1.365	0.987
800	96	100	12	2.524	13.366	0.906	1.305	0.988
1000	**120**	**100**	**12**	**2.157**	**9.940**	**0.847**	**1.200**	**0.990**
50	6	200	24	16.981	57.084	2.531	3.252	0.928
100	12	200	24	15.025	49.937	2.383	3.057	0.935
200	24	200	24	15.114	55.836	2.357	3.091	0.934
400	48	200	24	14.058	50.333	2.282	2.975	0.937
600	72	200	24	12.617	48.434	2.158	2.821	0.945
800	96	200	24	10.389	38.372	1.973	2.549	0.955
1000	**120**	**200**	**24**	**8.971**	**36.862**	**1.828**	**2.373**	**0.961**

Bold entries denote the lowest MSE and MAE values for a given output window size.

4.4.2 Performance of the CNN and Comparisons with LSTM Network

A CNN model was trained and compared with the LSTM network. The study specifically focused on using six and 120 input time-steps for the CNN, which are the smallest and largest input sizes used with the LSTM, to predict 1, 3, 6, 12, and 24 output time-steps. The performances of both models, measured in MAE, MSE, and Pearson

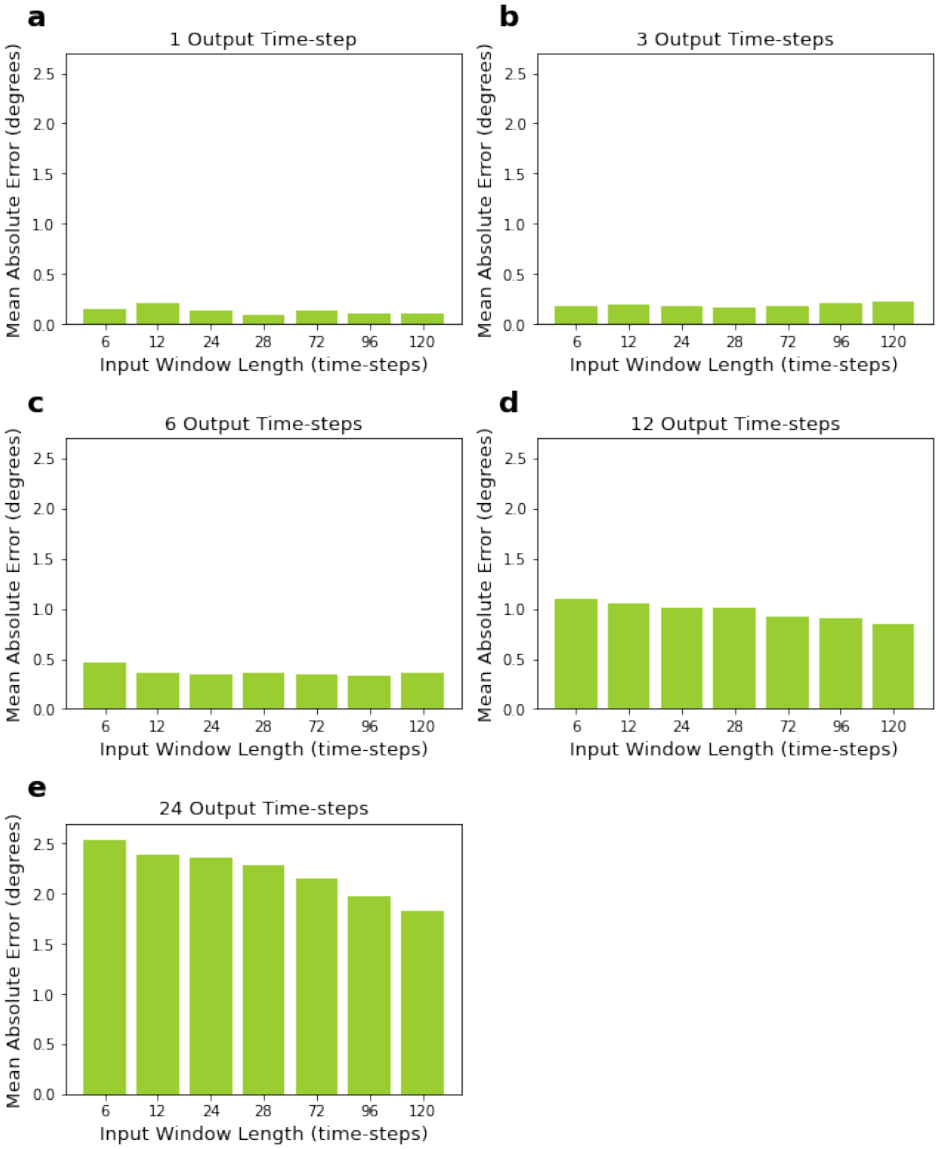

Figure 4.4: Performance of LSTM, measured by MAE, for varying input and output window sizes. Input sizes range from 6 to 120 time-steps, corresponding to 200 to 1000 ms. Output sizes range from 1 to 24 time-steps, corresponding to 8.33 to 200 ms. Sub-figures 4a, 4b, 4c, 4d, and 4e correspond to networks with 1, 3, 6, 12, and 24 output time-steps, respectively.

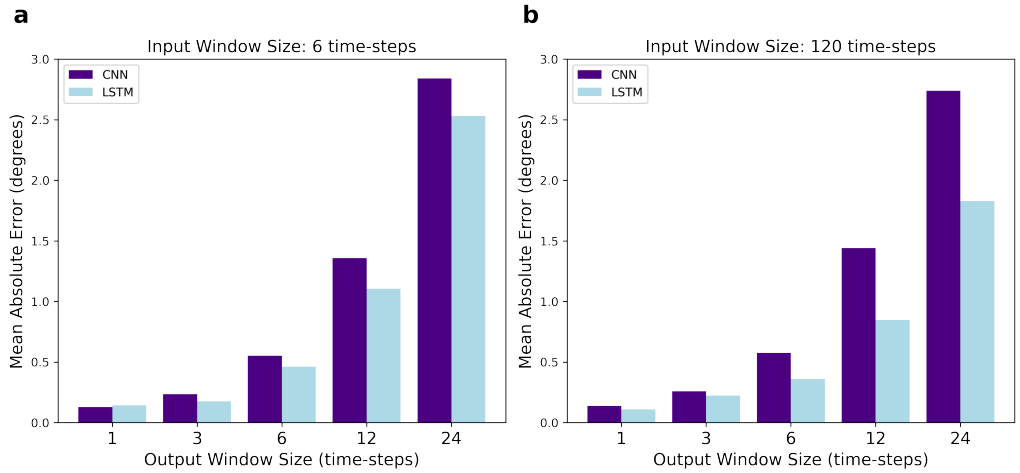

Figure 4.5: Comparison of the performance of the CNN and LSTM network, measured by MAE, for varying output window sizes. In (**a**), the input size is fixed at 6 time-steps, while in (**b**) the input is fixed at 120 time-steps.

correlation coefficient, are reported in Table 4.3.

Table 4.3: Performance of CNN in forecasting gait trajectories for varying input and output window sizes.

Input Window Size (ms)	Input Time-Steps	Output Window Size (ms)	Output Time-Steps	MSE (Degrees)	MSE std (Degrees)	MAE (Degrees)	MAE std (Degrees)	Mean Pearson Correlation Coefficient
50	6	8.33	1	0.069	0.960	0.129	0.229	1.000
50	6	25	3	0.184	1.699	0.234	0.360	0.999
50	6	50	6	0.891	4.685	0.552	0.766	0.996
50	6	100	12	5.265	20.277	1.358	1.850	0.977
50	6	200	24	20.437	65.596	2.840	3.517	0.913
1000	120	8.33	1	0.061	0.709	0.138	0.203	1.000
1000	120	25	3	0.216	2.020	0.259	0.386	0.999
1000	120	50	6	0.994	4.966	0.576	0.814	0.995
1000	120	100	12	5.496	18.724	1.440	1.850	0.975
1000	120	200	24	18.007	54.453	2.738	3.242	0.926

According to the results in Table 4.3, the errors of the CNN increase as the number of predicted time-steps increases, similar to what was observed with the LSTM. In Figure 4.5 we compare the performance of the CNN and LSTM, and it is evident that their mean errors are quite similar when predicting small output windows, such as one and three time-steps. However, as the output window sizes increase to 6, 12, and 24 future time-steps, the difference in the MAE between the CNN and LSTM becomes more pronounced, with LSTM outperforming CNN. Interestingly, the magnitude of this difference depends on the input window size: when six time-steps are used as input, the CNN MAE is larger than that of the LSTM; when 120 time-steps are used as input, the difference in their MAEs is even larger.

Table 4.4: Benchmarking performance of deep learning models

Input Window Size (ms)	Input Time-Steps	Output Window Size (ms)	Output Time-Steps	LSTM MAE (Degrees)	CNN MAE (Degrees)	FCN MAE (Degrees)	Naïve Method 1 [a] MAE (Degrees)	Naïve Method 2 [b] MAE (Degrees)
50	6	8.33	1	0.143 *	**0.129** *,†	0.195 *,†	0.449 †	1.513 *,†
50	6	25	3	**0.175**	0.234 *,†	0.294 *,†	0.888 †	1.916 *,†
50	6	50	6	**0.461** *	0.552 *,†	0.568 *,†	1.517 †	2.486 *,†
50	6	100	12	**1.104** *	1.358 *,†	1.336 *,†	2.640 †	3.494 *,†
50	6	200	24	2.531 *	2.840 *,†	**2.489** *,†	4.417 †	5.096 *,†
1000	120	8.33	1	**0.109** *	0.138 *,†	0.369 *,†	0.448 †	6.090 *,†
1000	120	25	3	**0.223** *	0.259 *,†	0.594 *,†	0.888 †	6.121 *,†
1000	120	50	6	**0.359** *	0.576 *,†	0.902 *,†	1.520 †	6.167 *,†
1000	120	100	12	**0.847** *	1.440 *,†	1.454 *,†	2.651 †	6.254 *,†
1000	120	200	24	**1.828** *	2.738 *,†	2.320 *,†	4.448 †	6.385 *,†

[a] Naïve Method 1: all output time-steps are predicted to be the value of the last input time-step.
[b] Naïve Method 2: all output time-steps are predicted to be the mean value of the input time-steps.
* and † represent statistical significance compared to Naïve Method 1 and LSTM, respectively. Significance is based on pairwise t-tests (p <0.05). Bold entries denote the lowest MAE value for a given input and output window size.

4.4.3 Benchmarking Performance of Deep Learning Models

A benchmark comparison between the performance of the two deep learning models and between a simpler machine learning algorithm, the Fully Connected Network (FCN), and two naïve/non-intelligent methods was conducted. The first naïve method predicted the final time-step in the input window as the value for all output time-steps, while the second naïve method used the mean value of the input as the predicted value for all output time-steps. Table 4.4 displays the MAEs for the deep learning and benchmark models. The deep learning models outperformed all naïve methods and the majority of the FCN predictions. However, there were two instances where the FCN obtained better results. The first was when the input and output window sizes were six and 24 time-steps, respectively, where the FCN had lower MAE compared to the LSTM and CNN. The other case was when the input and output sizes were 120 and 24 time-steps, respectively, with FCN performing better than CNN, but not better than LSTM.

Naïve Method 1 resulted in lower MAEs compared to Naïve Method 2, therefore, pairwise t-tests were conducted between the Naïve Method 1 and all other intelligent methods (LSTM, CNN, and FCN). This was done to determine whether the mean absolute errors were significantly different (with p <0.05); the results in Table 4.4 confirm that intelligent models perform better than non-intelligent models. Furthermore, pairwise t-tests were conducted between the LSTM and all the other models; the differences in the MAEs were found to be statistically significant, as indicated in Table 4.4.

4.4.4 Accuracy of the Models across the Different Time-Steps

In the previous subsections of this chapter, the mean errors across all features and time-steps were recorded. In this subsection, the MAE is calculated separately for each time-step for a given output window (see Figure 4.6). The MAE is calculated using an adapted form of Equation 4.7 (see Section 4.3.7), where the summation over $k = 1{:}l_{out}$ is not performed. As expected, the results show that predictions further in the future tend to deviate more from the actual values. This deviation becomes more pronounced after approximately the 3rd time-step, as shown in Figure 4.7. Figure 4.7 also shows that the LSTM MAE increases with larger output window sizes.

4.4.5 Performance of the Models for Each Joint

The association of higher errors with a particular type of joint was investigated. The MAE results for each of the hip, knee, and ankle joints are presented in Figure 4.8. The MAE for each joint represents the combined errors for the angles predicted in the pitch, roll, and yaw dimensions. They were calculated using an adapted form of Equation 4.7 (see Section 4.3.7), where the numbers of features, f, in the summation over $j = 1{:}f$ are reduced to the pitch, yaw, and roll angles for a single joint, rather than for all joints. However, the results did not reveal any specific trend indicating consistently higher errors for a particular joint.

4.5 Discussion

In this chapter, deep learning models were implemented, specifically LSTM and CNN, to forecast trajectories of children with pathological gait. This is the first time deep learning models have been applied for predicting trajectories of pathological gait of children, which exhibit larger inter- and intra-subject variability compared to the trajectories of typically developing children.

The advantage of using deep learning models is that they make predictions based on current input data while leveraging the learned representations of gait trajectories

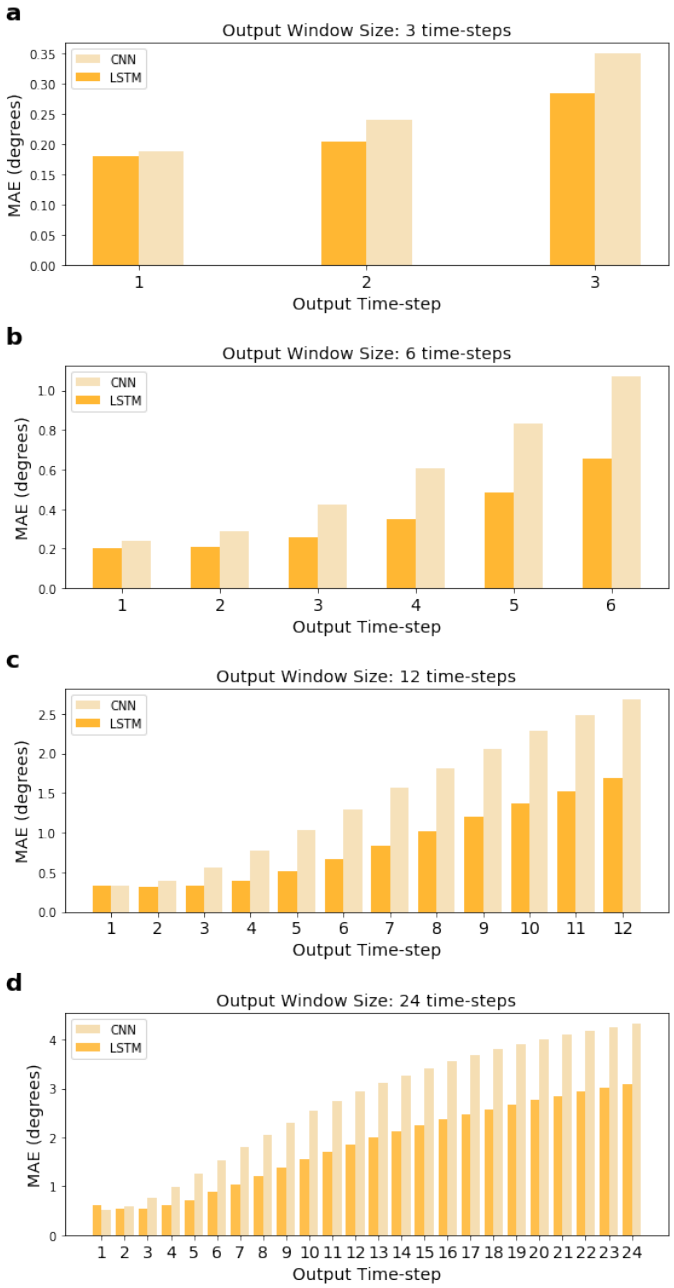

Figure 4.6: Mean absolute errors for each individual time-step predicted by the LSTM and CNN for a given output window. Input window size is fixed at 120 time-steps. Sub-figures 6a, 6b, 6c, and 6d correspond to networks with 3, 6, 12, and 24 output time-steps, respectively.

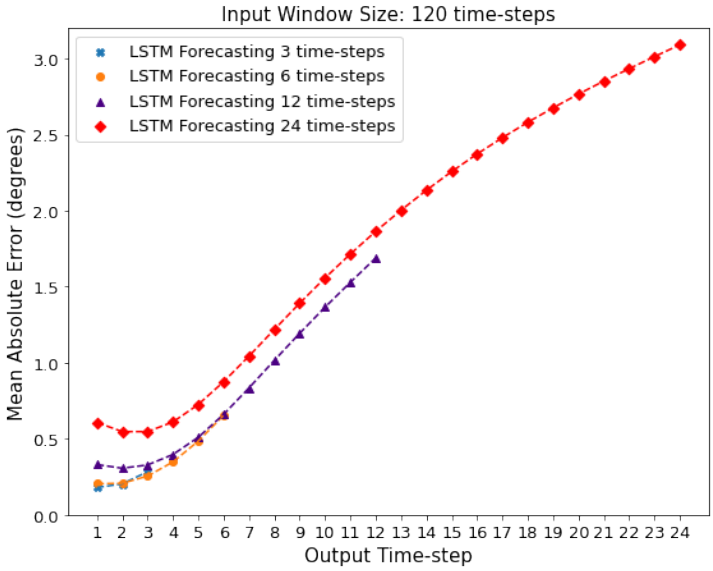

Figure 4.7: Mean absolute errors for each individual time-step predicted by the LSTM network with 3, 6, 12, and 24 output window sizes. Input window size is fixed at 120 time-steps.

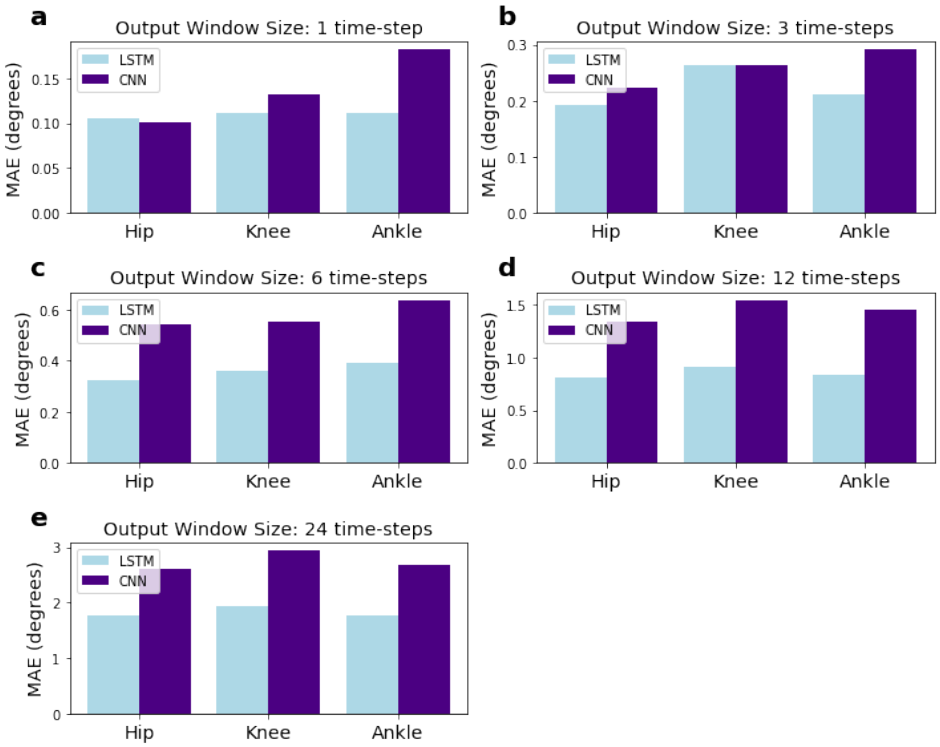

Figure 4.8: MAEs for each of the hip, knee, and ankle joints for the CNN and LSTM networks with varying output sizes. Input window size is fixed at 120 time-steps and the MAE for each joint represents the combined MAE for the yaw, pitch, and roll dimensions. Sub-figures 8a, 8b, 8c, 8d, and 8e correspond to networks with 1, 3, 6, 12, and 24 output time-steps, respectively.

acquired during a prior learning stage from numerous gait sequences. The LSTM and CNN networks were utilised to forecast hip, knee, and ankle trajectories based on varying input and output window sizes. Input window sizes for the LSTM were 50, 100, 200, 400, 600, 800, and 1000 ms (for data captured at a sampling frequency of 120Hz, these durations correspond to 6, 12, 24, 48, 72, 96, and 120 time-steps). Input window sizes for the CNN were 50 and 1000 ms (corresponding to six and 120 time-steps). The reason behind using input window sizes up to 1000 ms is to cover approximately one full gait cycle or lower as the average length of a gait cycle for a typically developing school-aged child is 980–990 ms [143]. The output window sizes for the LSTM and CNN were 8.33, 25, 50, 100, and 200 ms (corresponding to 1, 3, 6, 12, and 24 time-steps) meaning that up to 20% of the cycle could be predicted with those models. The time ranges used are based on values proposed in the literature by researchers that forecasted healthy gait (refer to Section 4.3.2 for details).

The LSTM, a type of gated recurrent network known for its effectiveness with sequential data was used for forecasting gait trajectories [141]. The LSTM has the advantage of taking into account the order of values in an input sequence and has the ability to learn long-term dependencies [141]. The LSTM network was compared to a CNN, mostly used for computer vision problems with 2D grid-like topology inputs using 2D convolutions, but it is increasingly employed with time-series sequences using 1D convolutions [144]. The CNN was implemented to evaluate if it shows promising performance in the task of forecasting trajectories for children with neurological disorders.

The results showed that the LSTM's performance is better than the CNN, particularly with larger input and output window sizes. The difference in MAE between the two networks was highest when using 120 time-steps as input, and 24 time-steps as output, with a gap of 0.91 degrees. There was one instance where the MAE for the CNN was higher than the LSTM, which occurred with the smallest combination of input and output windows (six and one time-steps, respectively). However, the difference in MAE between the two networks was small (0.014 degrees), and the CNN had a larger standard deviation. These findings differ from the results of other studies such as one by Moreira et al. [145], who found that the CNN was more robust for ankle joint torque estimation based on kinematics, speed, and anthropometry. The findings are

also different from the results of Molinaro et al. [146], where their temporal convolution network (with dilated convolution layers) outperformed an LSTM network. Its worth noting that Molinaro et al. considered joint moments rather than joint angles.

The influence of the size of the input and output windows on predictions was also investigated in this study. The size of the input window did not have a significant influence on the accuracy of the LSTM network when predicting small output window sizes (including one, three, and six time-steps). However, for longer output window sizes (including 12 and 24 time-steps), larger input windows resulted in lower errors. For both 12 and 24 time-steps, the lowest error was achieved using 120 input time-steps. This finding contrasts with a previous study by Zaroug et al. [132], who reported that increasing the input size beyond 30 time-steps led to increased mean errors when predicting five time-steps in the future (10 time-steps correspond to 60 ms in that study).

There were cases in the results of this study where the difference between the actual and predicted trajectories was large compared to the mean absolute error. It's difficult to determine whether this was due to the model's lack of generalisability for certain types of pathological gait patterns, or related to an underlying issue with the data for those samples, such as sensor or marker errors. This limitation stemmed from the anonymisation of the subjects in the dataset which did not contain supplementary information/videos for each trial. Another limitation of this study, also stemming from the anonymisation of the dataset, was the inability to test whether there is a significant difference in performance between individualised models (models that are subject-specific and need to be trained on data from the user of the exoskeleton) and generalised models (models that are subject-independent and make predictions without the need to be trained on data from a specific user). This is an important consideration when designing exoskeleton control strategies.

4.6 Conclusions

To conclude, two deep learning models, LSTM and CNN, were used to forecast the trajectories of children with neurological disorders. The results demonstrated that our deep learning models outperformed the three baseline methods, with the LSTM being the top performer, except for two cases where the FCN performed better (see Section 4.4.5 for details). The variation of input and output windows was explored to enhance our understanding of the impact of the length of input data and the length of the future horizon on performance. A potential application of this presented approach is the control of lower limb robotics, whereby forecasted trajectories can serve as a proxy for the user's intentions. These intentions can be integrated into the control hierarchy of exoskeletons, specifically into the high-level control responsible for detecting the user's intention and passing it on to the mid and low-levels to generate appropriate movement commands. It is important to strike a balance between performance and speed in real-time systems, ensuring input windows are large enough to achieve acceptable errors without significantly slowing down the system. Building on the work conducted in this study, the performance of the models should be evaluated on data collected using wearable sensors (e.g., IMUs and foot pressure sensors) rather than motion capture systems. This will be addressed in chapter 5. The difference between the performance of individualised and generalised models should be also studied. Furthermore, forecasted trajectories should be coupled with a corrective algorithm, tailored to each gait sequence. In this scenario, the user intention derived from a trajectory forecasting model would be adjusted by a corrective algorithm that produces the "desired trajectory" used by the mid- and low-level controllers of the exoskeleton. This has been addressed in the following study, presented in chapter 5.

Chapter 5: Reference Trajectory Generation for Children with Cerebral Palsy

5.1 Overview

This chapter presents the results of gait trajectory prediction of two distinct populations: typically developing children and children with Cerebral palsy. To accomplish this, four state-of-the-art deep learning models, namely a transformer, long-short-term memory network, convolutional neural network, and a fully connected neural network are implemented. The predictive performance of these models is evaluated in short-term and long-term (recursive) predictions and in the presence of varying levels of added Gaussian noise. This study introduces a methodology aimed at enhancing the stability of long-term prediction of joint kinematics. Furthermore, this chapter proposes a methodology for generating individualised, adaptive, and continuous trajectories tailored for children with Cerebral palsy, that can serve as reference trajectories for position-controlled exoskeletons.

5.2 Introduction

There are exoskeletons that have been specifically developed for children with CP [11], and Sarajchi *et al.* present a comprehensive literature review on this topic [11]. There are approximately fifteen single-joint and multi-joint lower limb exoskeletons primarily designed for children with Cerebral Palsy [11], including HAL [147], P-LEGS [148],

Trexo [149], CPWalker [150], EExRoLEG [151] and WAKE-up [152].

Incorporating knowledge of future gait trajectories can enhance the performance of the exoskeleton, by providing feed-forward input to the low-level controllers rather than utilising feed-back input only [97]. This approach can improve tracking of the exoskeleton's movement, and compensate for the control time-delays [97]. Several probabilistic and machine learning-based methods have been used to predict future gait trajectories [19, 97, 135, 137, 153]. However, these methods still need to be evaluated for stability in their predictions, which can be impacted due to measurement or controller noise, as well as during signal acquisition and transmission.

Furthermore, many exoskeletons currently follow a fixed gait trajectory, often based on the mean trajectory of a healthy population [154]. However, this may not be the most suited trajectory for the user, since it may not take into account their specific parameters, such as height and limb length, which have been shown to influence gait [155]. Some studies have focused on generating normalised gait cycles based on individual body parameters [35, 36]. While this approach provides more personalised gait trajectories to follow, it does not consider stride-to-stride variability during gait or the asymmetry between the left and right joints.

Motivated by the current limitations, several novel contributions are presented in this chapter. Firstly, stable deep-learning models are developed to predict one-step-ahead kinematic trajectories, specifically flexion-extension angles of the hip, knee, and ankle joints of both legs. A methodology is presented to optimise the long-term stability of these models by using dynamic time warping (DTW) distance metric for early stopping during training. The stability of the models was evaluated through (1) recursive forecasting (where predictions are used as input to the models, leading to the propagation of errors), and (2) the addition of varying levels of Gaussian noise to the input of the models (1-5%). The performance of the models in predicting the gait patterns of TD and CP individuals is assessed. Finally, an approach is proposed for generating continuous individualised and corrective reference trajectories for children with CP, taking into account the stride-to-stride variability and asymmetry of gait. This approach involves training the deep learning models on gait from typically developing children only, feed-

ing the models with CP gait as input, and then using the predictions from the trained models as potential reference trajectories for exoskeletons. It is hypothesised that these models can learn features of 'healthy' gait patterns. When CP gait patterns are used as input, these models can 'correct' CP patterns by introducing TD gait patterns, while still considering the individual features of the child and the asymmetry of their gait. The focus of this study is on predicting pediatric gait patterns and implementing the above using a variety of deep learning models including long-short-term-memory (LSTM), fully connected network (FCN), convolutional neural network (CNN), as well as Transformers which have never been investigated for gait trajectory prediction.

5.3 Background

Forecasting gait trajectories is valuable in exoskeleton control for several purposes. The predicted future trajectories can be used as feed-forward input to the controllers, enabling better tracking of the exoskeleton's movement [97], and compensating for delays in controller response times [97, 133]. Future trajectories can serve as target trajectories, as a guide for users to follow [154].

Several approaches have been used for the gait trajectory forecasting task, including probabilistic models [97] and deep learning models such as LSTMs and CNNs [135,137]. These approaches vary in the number of time-steps predicted, ranging from single to multiple time-steps in the future. The predicted trajectories can take the form of joint angles, linear accelerations or angular velocities. Another difference amongst the approaches is the input parameter or sensors used to collect the data to develop the models which included motion capture systems [132, 136, 137, 156], IMUs [133, 134], encoders [131], and surface electromyography (sEMG) [135, 157]. A summary of some approaches in literature is provided in Table 5.1. Real-time predictive models are prone to receive noisy inputs, during signal acquisition and transmission. Therefore, it is crucial for models predicting gait trajectories intended for exoskeleton control to be robust to noise and exhibit stability in their predictions. While the accuracy of these models has already been evaluated, the stability of these models is yet to be investigated.

Table 5.1: Summary of the results of the papers that implement trajectory prediction

Paper	Method	Predicted Parameter	Sensors / Modality	Input window size	Output Window size	Error Metric	Error	Subjects
Jia *et al.* [135]	LSTM with feature level-fusion	Knee angle	Motion capture and EMG	-	50ms	RMSE	0.464 ± 0.096	4 healthy
Zhu *et al.* [137]	Attention based CNN-LSTM	Knee and ankle angle	Motion capture	-	60ms / 9 time-steps (150 Hz)	RMSE / MAE	0.317 / 0.202	Healthy
Zhu *et al.* [137]	CNN-LSTM	Knee and ankle angle	Motion capture	-	60ms / 9 time-steps (150 Hz)	RMSE / MAE	0.665 / 0.523	Healthy
Zhu *et al.* [137]	LSTM	Knee and ankle angle	Motion capture	-	60ms / 9 time-steps (150 Hz)	RMSE / MAE	0.745 / 0.562	Healthy
Zhu *et al.* [137]	CNN	Knee and ankle angle	Motion capture	-	60ms / 9 time-steps (150 Hz)	RMSE / MAE	0.619 / 0.810	Healthy
Ling *et al.* [157]	Back Propagation Neural Network	Knee angle	Surface electromyograph (sEMG)	-	1 time-step	RMSE	4.406 – 5.945 (offline) / 5.219-6.622 (online)	4 healthy
Ling *et al.* [157]	LSTM	Knee angle	Surface electromyograph (sEMG)	-	1 time-step	RMSE	6.3600-12.2033 (offline) / 6.4928-12.5539 (online)	4 healthy
Ling *et al.* [157]	SVR	Knee angle	Surface electromyograph (sEMG)	-	1 time-step	RMSE	8.2800-10.0430 (offline) /8.7688-10.6569 (online)	4 healthy
Liu *et al.* [81]	Deep Spatial temporal model	Hip and knee angle	Encoder	50 time-steps (50Hz)	1 time-step (50Hz)	-	Refer to fig 6 in paper	35 healthy (25±3)
Hernandez *et al.* [134]	Deep Conv LSTM	Hip, knee ankle, lumbar	IMU	100 time-steps (100 Hz)	100 time-steps (100 Hz)	MAE	3.6 ± 2.1 (average)	27 healthy (26.5 ± 3.9 years)
Moosavian *et al.* [158]	LSTM	Robowalk hip and knee angle	-	64 (step size of 3)	-	-	5 degrees (maximum), 0.5 degrees for stepping up mode	7 (5 healthy + 2 transfemoral prosthesis users)
Ren *et al.* [156]	LSTM	Hip, knee, ankle	Motion Capture	-	1 time-step	RMSE	3.738-18.052 (Conventional LSTM) / 0.392-0.778 (LSTM with real time updates)	CMU database
Zaroug *et al.* [132]	LSTM	Linear acceleration and angular velocity	Motion Capture	25 time-steps (100Hz)	5 time-steps	MAE	0.047 m/s2 thigh LA, 0.047 m/s2 shank LA, 0.028 deg/s thigh AV, 0.024 deg/s shank AV.	6 healthy male (age 22 ± 2 years)
Zaroug *et al.* [136]	Vanilla LSTM	Linear acceleration and angular velocity	Motion Capture	25 time-steps / 0.5 sec (50Hz)	5 time-steps / 0.1 sec (50Hz)	MAE	0.221-0.379	13 healthy male and 3 female (28 ± 4years)
Zaroug *et al.* [136]	Stacked LSTM	Linear acceleration and angular velocity	Motion Capture	25 time-steps / 0.5 sec (50Hz)	5 time-steps / 0.1 sec (50Hz)	MAE	0.176-0.340	13 healthy male and 3 female (28 ± 4years)
Zaroug *et al.* [136]	Bi-LSTM	Linear acceleration and angular velocity	Motion Capture	25 time-steps / 0.5 sec (50Hz)	5 time-steps / 0.1 sec (50Hz)	MAE	0.204-0.365	13 healthy male and 3 female (28 ± 4years)
Zaroug *et al.* [136]	ED-LSTM	Linear acceleration and angular velocity	Motion Capture	25 time-steps / 0.5 sec (50Hz)	5 time-steps / 0.1 sec (50Hz)	MAE	0.176-0.336	13 healthy male and 3 female (28 ± 4years)
Su *et al.* [133]	LSTM	Angular velocity	IMU	10 time-steps or 30 time-steps	10 time-steps / 200ms or 5 time-steps / 100ms (50Hz)	MAE	0.063-0.308	12 healthy (6 males and 6 females) (25-30 years)

In addition to using predictive model outputs as feed-forward to controllers, predicted gait trajectories can be used as reference trajectories for exoskeletons [154]. Exoskeletons that operate based on position control rely on a reference trajectory, often derived from healthy individuals, to dictate the desired joint positions during a gait cycle. This reference trajectory is used to correct pathological gait patterns [117], but it does not take into consideration several parameters that influence gait, including speed, gender, and anthropometrics [155]. Gaussian process regression and recurrent neural networks (RNNs) have been used to address this issue, by generating healthy gait trajectories based on parameters such as speed, gender and anthropemtrics [35]. These models learn the mapping between body parameters and healthy gait cycles, allowing the generation of personalised reference trajectories for each individual. Using individualised gait trajectories for gait rehabilitation has resulted in improvements in energy efficiency, measured by an increase in heart rate and reduction in peripheral capillary oxygen saturation (SpO2), compared to generalised trajectories [159].

While these approaches provide more individualised trajectories, they are fixed for all gait cycles and do not consider the inherent cycle-to-cycle variability observed during gait. Children with spastic CP, have been shown to exhibit higher within-day and between-day variability in comparison to typically developing children, which can be due to the limited range of motion caused by their spasticity [139,140]. An online adaptive trajectory generation is needed to account for this cycle-to-cycle variability. Vallery *et al.* [117] used complementary limb motion estimation (CLME) for hemiparetic individuals, that relies on the trajectories of the healthy leg for the online estimation of the reference trajectory for the pathological leg. Using CLME approach was more efficient, led to EMG patterns that were closer to unperturbed gait than when using a fixed reference trajectory and avoided out-of-phase walking, which can be generated by using a fixed reference trajectory [117]. Nevertheless, this approach is specific to individuals with hemiparesis and not those who have both limbs affected. Meanwhile, Zhou *et al.* [36] used RNNs to generate normalised gait trajectories based on anthropometrics, as well as gait speed, yet their approach does not account for the kinematic asymmetry between the left and right joints.

These limitations were the motivation behind this study which presents one-step-ahead

kinematic trajectory prediction models that are optimised for stability in their long-term predictions and are evaluated for their robustness in the presence of added noise. This study also presents an approach to generate adaptive target/reference trajectories for children with CP, that vary from cycle-to-cycle, and take into account the asymmetry of the left and right joints, since a separate trajectory will be generated for each joint of the left and right sides. A similar approach has been done by Endo *et al.* [160], who trained a GaitForMer network based on healthy gait patterns for human motion forecasting, and then retrained the model that learned gait mappings to predict the severity of gait impairment of patients with Parkinson's disease, based on the Movement Disorder Society Unified Parkinson's Disease Rating Scale (MDS-UPDRS).

In this study, a transformer network has been implemented for gait trajectory prediction. Transformers have gained significant popularity, outperforming CNNs and LSTMs in several applications, as shown in [161]. Using transformers offers several advantages. Firstly, transformers rely on the attention mechanism rather than on recurrence or convolutions allowing for the capture of long-range dependencies. Moreover, the elements of the input sequence to a transformer are processed in parallel, enabling faster training [162]. The promising results of transformers demonstrated in literature have inspired its use in this study [161]. To the best of our knowledge, this is the first time transformers have been utilised for gait trajectory prediction.

5.4 Methodology

5.4.1 Overview

This study involves developing end-to-end kinematic trajectory prediction models, that perform one-step-ahead prediction of joint angles of the hip, knee and ankle for both legs. The models are trained using 100 time-points of past joint angles (equivalent to 1000ms for a sampling frequency of 100Hz) and optimised for long-term stability in their predictions by using dynamic time warping distances (DTW), in addition to validation loss, as metrics to end training of the models. Four deep learning models are implemented, including LSTM, FCN, CNN and transformer that, importantly, are

trained on the gaits of typically developing children. Their stability is evaluated in long-term forecasting of 200 time-steps in the future, which is twice the length of the input size. The stability of the four models is evaluated by: (1) performing recursive prediction which can lead to the propagation of errors, and (2) by adding varying levels of Gaussian noise to the input (1-5%). Finally, the four models are used to predict future trajectories (200 time-steps ahead) when using 100 time points of CP gait as input. It is hypothesised that the predictions from models that learned the gait patterns of typically developing children could be used as an aid to correct the gait of children with CP.

5.4.2 Data

The deep learning models implemented in this study were trained and evaluated on the Canterbury Christ Church University and Chailey Clinic Dataset, which contains recordings of gaits from both typically developing children and children with Cerebral Palsy. The dataset includes flexion-extension angles of the hip, knee and ankle measured simultaneously in the sagittal plane, for the right and left legs. Details on this dataset can be found in Chapter 3, section 3.2.2.

5.4.3 Pre-processing

The data of typically developing children were divided into 3 subsets: training, validation and testing sets. The division was performed at the subject level (i.e. data from 8 children were used for training, data from the other 2 children for testing and data from 1 child for validation). The children for each set were selected at random. This was done to avoid testing the models on samples from a child used for training and therefore ensure the generalisability of the models.

Prior to training the models, each one of these sets was pre-processed by segmenting the trials into samples; each trial is a recording of the flexion-extension angles of the hip, knee, and ankle joints while walking for 8 meters. Each sample consisted of an input matrix x_{in}, which contained joint angle values for 100 time-steps, and a target

vector y_{out}, which contained the joint angle values for the subsequent time-step. The input window size was specifically chosen to be 100 time-steps because this corresponds to 1000ms (for a 100Hz sampling frequency), which is equivalent to the length of one full gait cycle as the average length of a gait cycle for TD school-aged children is 980-990ms [143]. This allows the model to make predictions based on one full previous cycle of an exoskeleton user. In chapter 4, the effect of varying the length of the input window on the accuracy of predicting trajectories in the form of Euler angles was investigated [153]. The range of input window sizes used in that study were 50, 100, 200, 400, 600, 800, and 1000 ms. Results showed that for short-term predictions, the size of the input window does not have a significant influence on accuracy, while for long-term predictions, larger input window sizes result in better performance. This further supports our choice to set the input window size to 100 time-steps (equivalent to 1000ms). The decision to set the input window size to 100 time-steps (equivalent to 1000ms) for this study was based on these findings.

For n samples in a set, $X_{in} \in \mathbb{R}^{n \times l_{in} \times f}$, where l_{in} (set to 100) is the number of input time-steps and f (set to 6) is the number of features that we input to the models (hip, knee and ankle angles for the left and right leg). Similarly, $Y_{out} \in \mathbb{R}^{n \times l_{out} \times f}$, where n is the number of samples, l_{out} (set to 1) is the number of target time-steps, while f (set to 6) is the number of features. The samples were generated using the sliding window method (discussed in chapter 4) [153]. The stride value was set to 1 to maximize the number of training samples that can be generated from each trial. For the typically developing gait data, the training, testing, and validation sets consisted of 41120, 7316, and 4832 samples, respectively.

In addition to these 3 main sets, two additional long-term prediction validation and testing sets were generated. These sets had the same input size as in X_{in} (i.e. 100 time-steps), but they have an output size (l_{out}) of 200 time-steps. These sets were generated to evaluate the feasibility and stability of recursive long-term forecasts of the trained models (see section 5.4.5.2 and section 5.4.6 for further details).

All the data from children with CP were used *only* for testings the models, both for one-step-ahead prediction and long term predictions. The data from children with CP

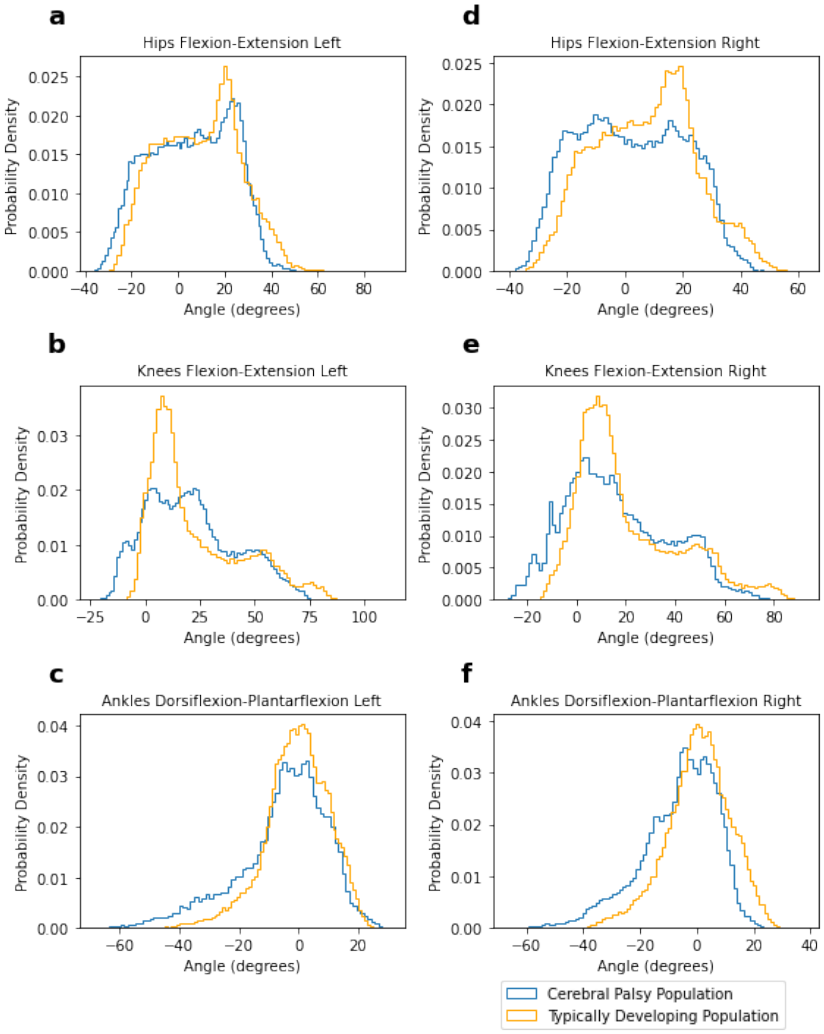

Figure 5.1: Probability density distribution of the hip, knee, and ankle angles (in degrees) for TD and CP gait data before processing. The blue and yellow lines correspond to CP and TD probability density distributions, respectively.

were processed in the same manner as the data for typically developing children.

After generating the sets, they were normalised using min-max normalization, such that $X_{in} \in [0, 1]$ and $Y_{out} \in [0, 1]$. The testing and validation sets are normalised according to the normalisation factors (i.e. min and max values) used during training. The min-max values were chosen to account for the joint angle ranges of both the typically developing and CP distributions with an additional safety boundary. This was done to ensure that the models are capable of handling test data from subjects that have slightly different joint angle ranges while keeping input bounded between 0 and 1. Also, this is to accommodate for differences in CP and TD gait data distributions which can be shown in Figure 5.1.

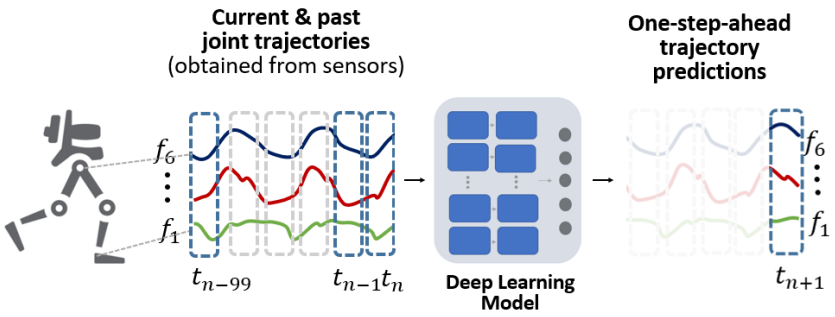

Figure 5.2: Illustration of one-step-ahead gait trajectory prediction models. Based on a 100 time-step window of six input features (f1-f6), the models make one-step-ahead-predictions for each feature. The features are the flexion-extension angles of the hip, knee, and ankle for both legs.

5.4.4 One-step-ahead trajectory prediction models

Four deep learning models were implemented for one step ahead prediction of gait trajectories, including a Fully Connected Neural Network (FCN), a Long Short-Term Memory (LSTM), a Convolutional Neural Network (CNN), and a Transformer. These sequence-to-sequence models were trained to predict one-step ahead gait trajectories based on a 100 time-step input window of joint angles, specifically the hip, knee and ankle angles of the left and right foot (see Figure 5.2). The models' $g(X)$ learns the mapping between input X (made of 100 time-steps) and the output $\hat{Y}$ (one-step ahead prediction), to minimise the difference between the estimated output $\hat{Y}$ and the true output Y. For n number of samples, l_{in} input window length, l_{out} output window length, and f features, the input of the model is matrix X, where $X \in \mathbb{R}^{n \times l_{in} \times f}$, and the output of the model is matrix $\hat{Y}$, where $\hat{Y} \in \mathbb{R}^{n \times l_{out} \times f}$. In the following subsections, the architecture of each of the models will be described.

5.4.4.1 Fully Connected Network (FCN)

The Fully Connected Network (FCN) consists of a series of fully connected linear layers with ReLU activation functions in between, and a final sigmoid layer as an output. The 2-dimensional input $\mathbb{R}^{100 \times 6}$ (given that we have 6 joint angles and a 100 time-step input window) has been flattened to a 1-dimensional vector $\mathbb{R}^{600}$ before passing it through the fully connected layers. A total of five linear layers were used, with the architecture

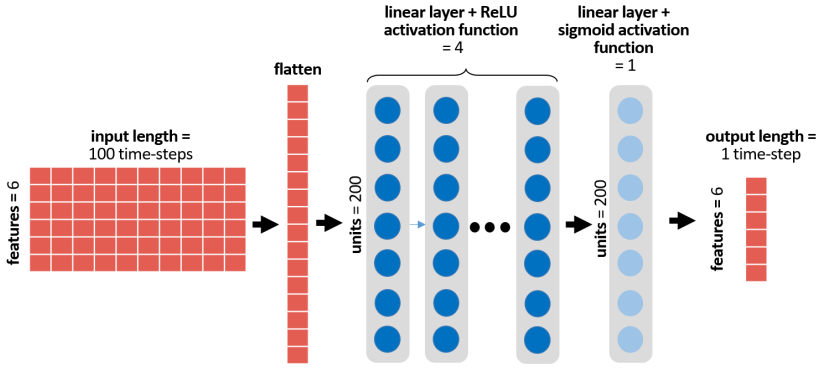

Figure 5.3: Architecture of the Fully Connected Neural Network (FCN). The network contains a total of five linear layers and predicts one future time step based on 100 input time steps.

shown in Figure 5.3. A note on how the FCN in this chapter differs from the ones implemented in Chapters 4 and 6 is included in Chapter 4, section 4.3.5.

5.4.4.2 Long-short-term-memory Network (LSTM)

The LSTM is a type of gated recurrent neural network commonly used with time-series data, since it processes the data sequentially. Each LSTM unit which consists of an input, output, and a forget gate, regulates the flow of information through the network. The parameters of these gates are set during the training process, to optimise performance [141]. For this study, a network that contains 2 layers, and 100 LSTM units per layer was implemented. The last hidden state of the final layer is then passed onto a fully connected layer before reshaping the output into the desired shape. The architecture of the LSTM network is shown in Figure 5.4.

5.4.4.3 Convolutional Neural Network (CNN)

While CNNs are primarily associated with 2-dimensional inputs such as images, they can be applied to 1-dimensional sequences, whereby the 2D convolution operation is replaced with the 1D convolution operation [144]. The CNN architecture implemented in this study contained two pooling and four convolution layers, followed by a fully connected linear layer at the end. A ReLU activation function was used after each convolution layer. The architecture of the CNN model is depicted in Figure 5.5.

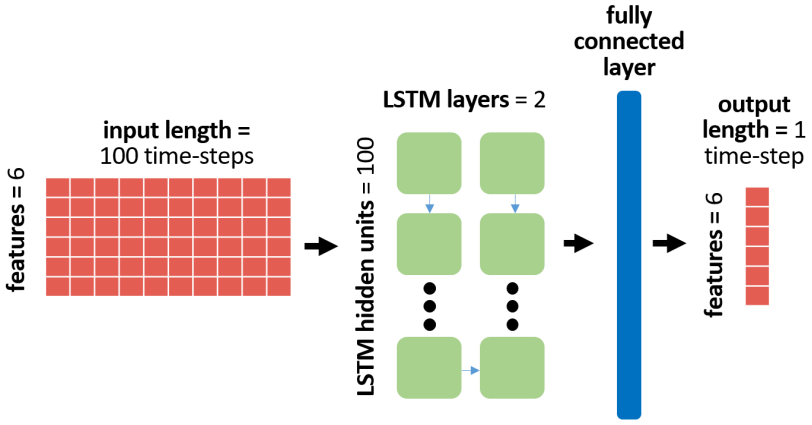

Figure 5.4: Architecture of the Long-short-term-memory (LSTM) Network. The network contains two LSTM layers with 100 hidden units each followed by one fully connected layer. The network predicts one future time step based on 100 input time steps.

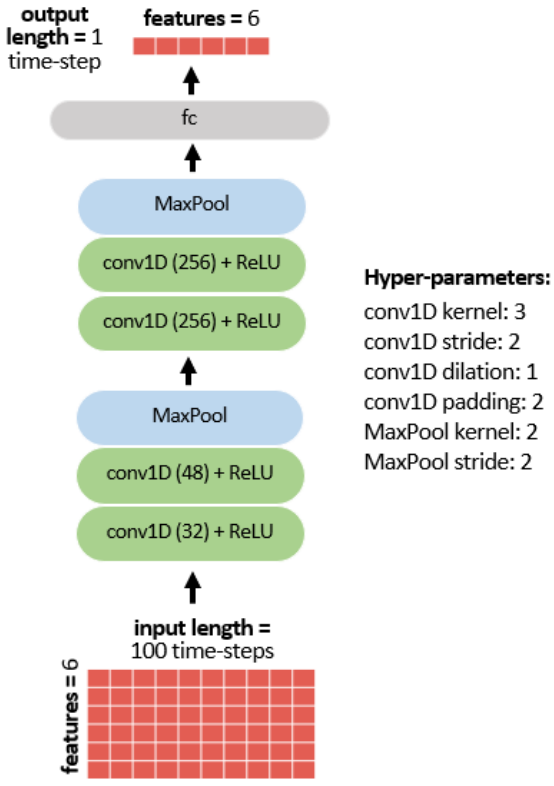

Figure 5.5: Architecture of the Convolutional Neural Network (CNN). The network contains a total of four convolutional layers, two pooling layers, and one fully connected layer. The network predicts one future time step based on 100 input time steps.

5.4.4.4 Transformer

The transformer architecture implemented for this study is based on the one proposed by Vaswani *et al.* [163]. The transformer consists of one encoder layer and one decoder layer. The input which consists of 100 time-steps of six joint angles (hip, knee, and ankle flexion-extension angles for both legs) is fed into a linear layer that expands the dimension from $\mathbb{R}^{100\times6}$ to $\mathbb{R}^{100\times80}$. The expansion of the input dimension is necessary to be able to set the number of multi-dimensional heads of the encoder model to 8. The output of the linear layer is concatenated with positional encodings, which are used to inform the model of the order of the sequence [163]. The result of the concatenation is fed into the encoder, which is a single layer consisting of 8 multi-attention heads, and a 100-unit feed-forward network. Meanwhile, the last time-step of the input is fed to a linear layer that expands the dimension from $\mathbb{R}^{1\times6}$ to $\mathbb{R}^{1\times80}$. The decoder receives two inputs: the output of the decoder's linear layer which is concatenated with positional encodings, and the output of the encoder (which is the output of the feedforward network added and normalised with a residual connection) [163]. The output of the decoder goes through a fully connected layer and then a sigmoid activation function. The positional encodings had a dropout rate of 0.2, while the encoder and decoder had a dropout rate of 0.1. The architecture of the Transformer implemented is illustrated in Figure 5.6.

5.4.5 Model Optimisation

The following subsections describe the training and optimisation processes for the LSTM, FCN, CNN and Transformer networks.

5.4.5.1 Hyper-parameters

All models were trained using the Adam optimiser, with the mean squared error (MSE) between one-step-ahead predictions and true values used as the loss function to update the weights of the models. The models were trained up to 40-50 epochs. We stored the models at the epoch where the DTW distance between the recursive predictions of

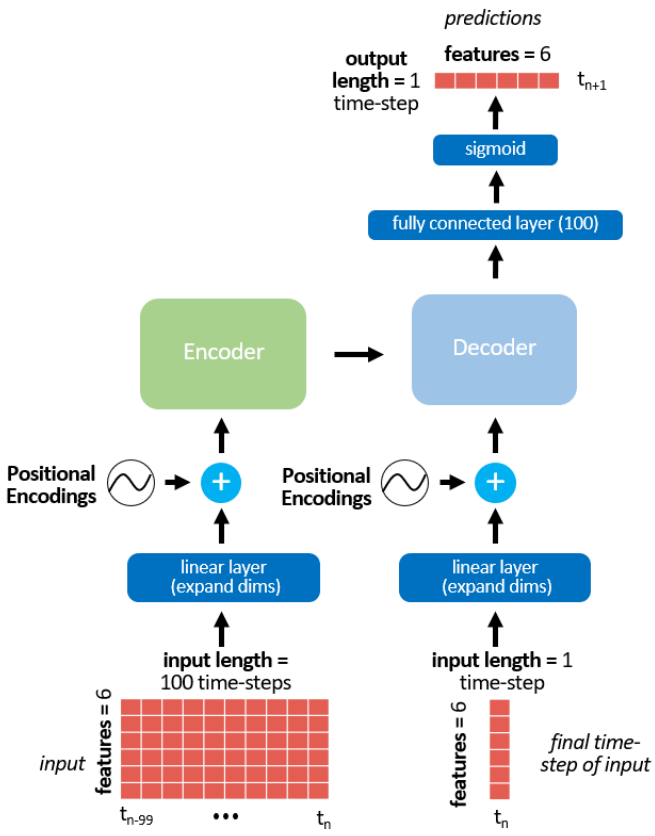

Figure 5.6: Architecture of the Transformer Network. The network consists of an encoder and a decoder and predicts one future time step based on 100 input time steps.

the 200 time-steps and the true joint angles of the validation set was the lowest. To select the optimal hyperparameters for the models in this study, we have started with hyperparameters that have been selected in the previous study, presented in chapter 4. Those hyperparameters were selected based on a hyperparameter search that uses the tree-structured Parzen estimator algorithm, a type of Bayesian hyperparameter sampler, and optimised for the prediction of trajectories in the form of Euler angles for children with CP. Details on the search space are included in chapter 4. Subsequently, these hyper-parameters have been fine-tuned to optimise the performance of the models on the current dataset. The batch sizes for the FCN, LSTM, CNN, and Transformer were 32, 256, 256, and 512 respectively. The learning rate was set at 0.0001 for the FCN, LSTM and CNN, and at 0.001 for the Transformer.

5.4.5.2 Dynamic Time Warping Distances as Early Stopping Criteria

In this study, the mean square error (MSE) between the one-step-ahead predictions and the true values was used as the loss function, with the models being trained to minimise the loss. While minimising this loss ensures low errors in short-term predictions, it does not guarantee that the models are not over-fit to short-term forecasting, and are capable of accurate long-term recursive forecasts. After each epoch of training, the DTW distance between the 200 recursively predicted time steps and the true gait values of the validation set is calculated. This distance serves as a measure of the model's performance in long-term forecasting. The validation loss was monitored to ensure the model is learning and performing well on short-term prediction, and the DTW distance on the validation set was used to determine when to end the training of the model to avoid overfitting and ensure the stability of the models in long-term forecasting. Figure 5.7 provides an example plot illustrating the training and validation MSE loss as well as the DTW distance measured for each epoch during the training of one of the models. The plot shows that initially, during the beginning of training, both the validation loss and DTW distance decreased, indicating improvement in short-term prediction and long-term recursive forecasting. However, after a certain number of epochs, the DTW distance increases, suggesting a degradation in the model's ability to accurately forecast long-term trajectories. Based on these observations, this training approach aims

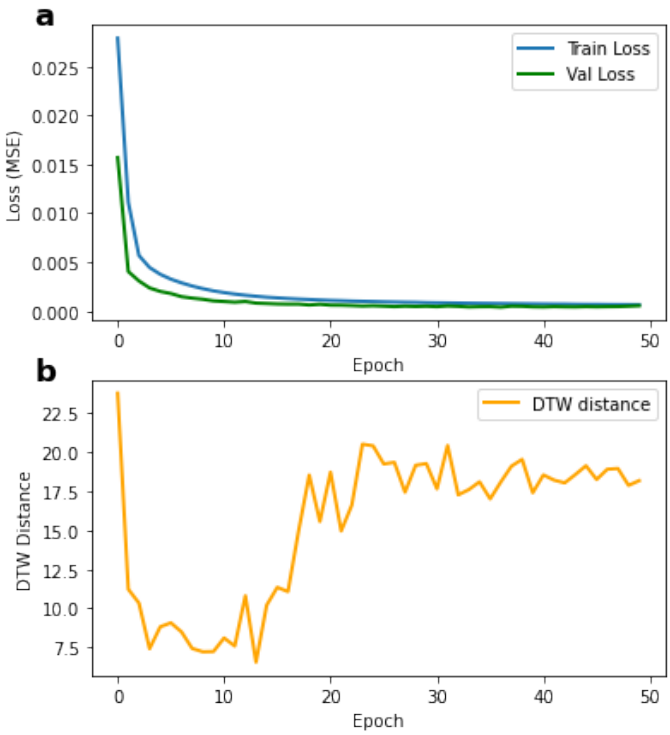

Figure 5.7: Plot of the loss curves during training of a model. (a) shows the training and validation MSE loss on one-step-ahead predictions, and (b) shows the dynamic time warping (DTW) distances between 200 recursively predicted outputs and the true outputs of the validation set.

to optimise for low one-step-ahead MSE validation loss while also minimising the DTW distance in long-term recursive forecasting. This strategy ensures that the models are capable of both accurate short-term predictions and reliable long-term forecasts. This approach is illustrated in Figure 5.8.

5.4.5.3 Framework

The Pytorch machine learning framework has been used to implement deep learning models. Several additional libraries were utilised including Numpy, Matplotlib, SciPy, Seaborn, and Scikit-learn. DTW python package was used for calculating dynamic time-warping distances [164]. Computation was run on an Nvidia Geforce RTX 2070 GPU.

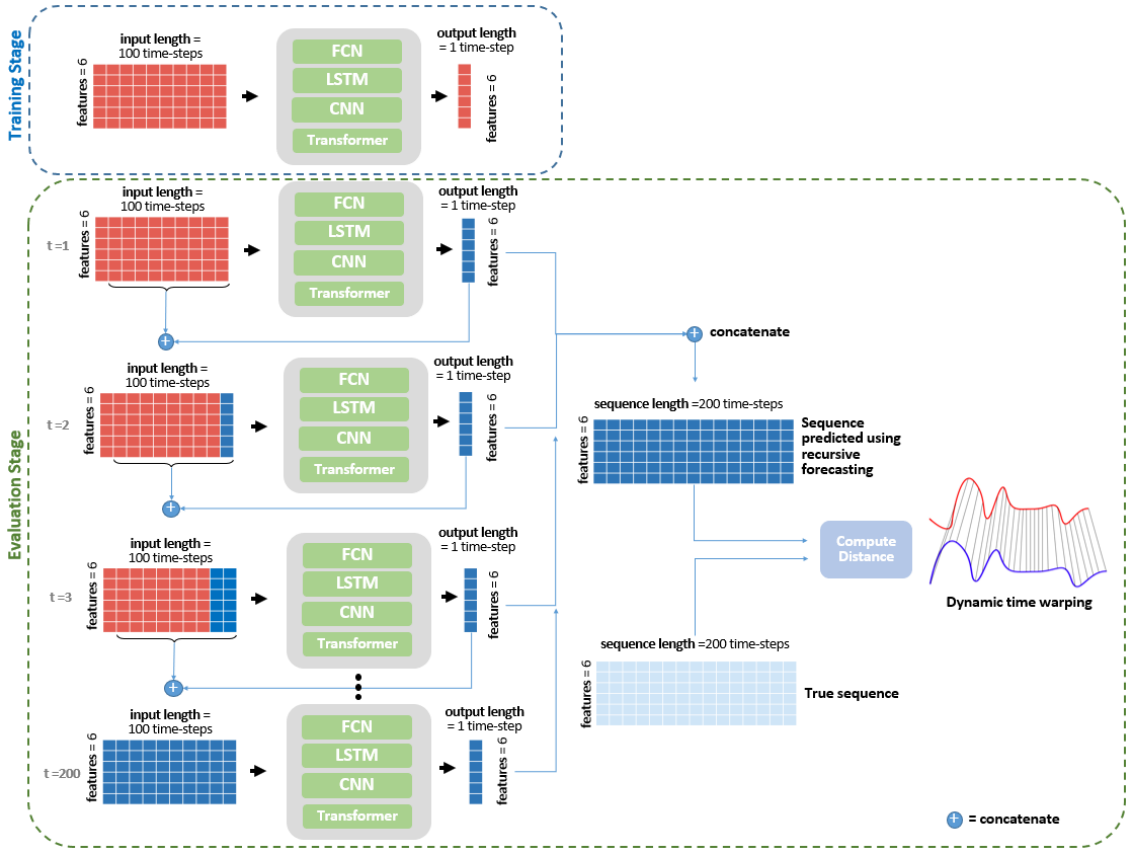

Figure 5.8: Strategy for optimising the stability of the models for the long-term gait prediction task. Dynamic time warping distances are calculated between 200 recursively predicted time-steps and the true gait values after each training epoch. This procedure is repeated after each epoch on the validation set. The weights that lead to the lowest DTW distances are saved for inference.

5.4.6 Long-term recursive trajectory forecasting

Recursive forecasting is a technique where the one-step-ahead predictions made by the model are reused as input for making subsequent predictions. In this study, this method has been used to evaluate the feasibility of long-term recursive forecasts (see section 5.4.7), and also as a metric for early stopping during training to optimise for long-term stability (see section 5.4.5.2). The one-step-ahead prediction models developed in section 5.4.4 were used for recursive forecasting.

5.4.7 Evaluating Stability

Two methods were used to evaluate the stability of the networks. The first method involved long-term recursive prediction. A stable network should be able to make accurate predictions over an extended period using recursive input, without being significantly affected by noise resulting from the propagation of error. The stability of the networks was evaluated by recursively predicting 200 time-steps into the future, equivalent to approximately two gait cycles, which is twice the length of input times-steps used by the model. During the first 100 recursive predictions, the input to the model consisted of a combination of true and predicted values. For the next 100 recursive predictions, the input was based solely on predicted values. The long-term predictions were compared to the true values and the errors between them were calculated to measure the network's stability.

The second method employed to assess the stability of the networks involved the addition of Gaussian noise to the predictions. This method was used in conjunction with the long-term recursive prediction described above and involved the addition of varying levels of Gaussian noise (1-5%) to each prediction before using it as input for the model. Subsequently, 200-time steps were recursively predicted into the future with added Gaussian noise, and the errors between these predictions and the true gait values were calculated.

5.4.8 Cerebral Palsy Gait Correction

Position-controlled exoskeletons often guide users to follow a reference/target trajectory, based on the mean trajectories of a healthy population, which results in corrections to their pathological gait patterns [117,154]. This study proposes an approach for generating personalized adaptive target/reference trajectories for individuals with CP by leveraging the one-step-ahead trajectory prediction models trained on the gait of TD children.

The process involves training one-step-ahead trajectory prediction models exclusively on TD gait. These models are then fed with CP gait, and the models' predictions are used as proposed reference/target trajectories for the child with CP (see Figure 5.9). The models make predictions based on input from both the right and left limbs, and produce separate output predictions for the right and left limbs, instead of the same trajectory for both limbs. This allows accommodating for the asymmetry and slight differences in right and left limb trajectories, especially for children with unilateral CP, where only one side is affected. We hypothesise that these models will learn the mappings of 'healthy' trajectories and the inter-joint couplings, and will introduce the TD patterns onto CP gait when CP gait is used as input.

Specifically, once the models have been trained on TD gait, the models are fed with 100 time-steps of CP gait and recursively predict 200 time-steps into the future. These predictions are compared with the natural evolution of CP gait to assess whether the models introduced TD patterns to the predictions.

5.4.9 Performance Metrics

For evaluating the predictive performance of the models in short-term (one-step-ahead) and long-term predictions, two common error metrics are used: mean squared error (MSE) and mean absolute error (MAE). These metrics were calculated after the denormalisation of the predictions (refer to Chapter 4, section 4.3.7 for the equations used to calculate MAE and MSE as well as their standard deviations).

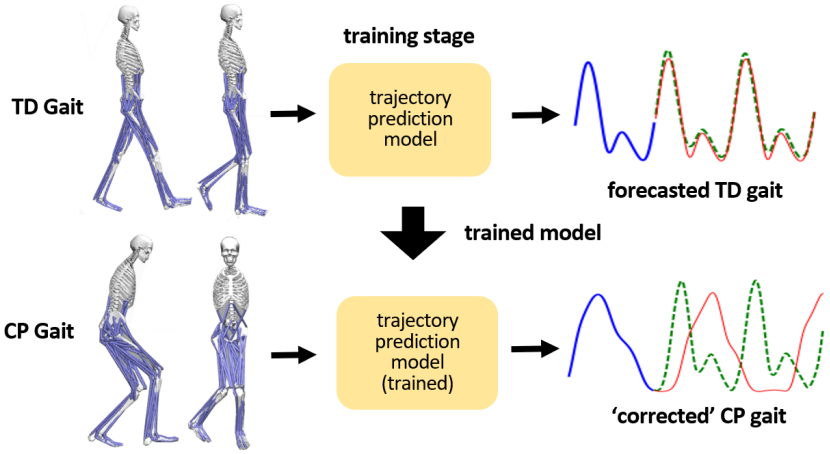

Figure 5.9: Methodology for generating adaptive reference/target trajectories for children with CP by training the deep learning models on the gait patterns of TD children. The blue lines correspond to the model's input, the green lines to the model predictions, and the red lines to the actual gait values.

5.5 Results

This section presents the results of the predictive performance of the four models, in the short-term (one-step-ahead, section 5.5.1) and the long-term (200 time-steps, section 5.5.2). It also reports the effect of noise on gait predictions (section 5.5.3), and showcases illustrative examples of gait trajectory corrections for children with CP (section 5.5.4).

5.5.1 Performance on short-term (one-step-ahead) predictions

Four deep learning networks (LSTM, FCN, CNN and Transformer) were trained for the task of one-step-ahead gait trajectory prediction (see Figure 5.10 and Figure 5.11). The input consisted of 100 time steps, representing hip, knee, and ankle angles in the sagittal plane for both legs. These networks were trained on the gait patterns of typically developing (TD) children. To evaluate the predictive performance of the models, a test set consisting of data from two TD children that were not included in the training process is used. The mean squared errors (MSE) and mean absolute errors (MAE) between the predictions and true values were calculated. Additionally, these models, trained on TD gait, are tested on data from 11 children with Cerebral Palsy (CP). The results of these evaluations are reported in Table 5.2.

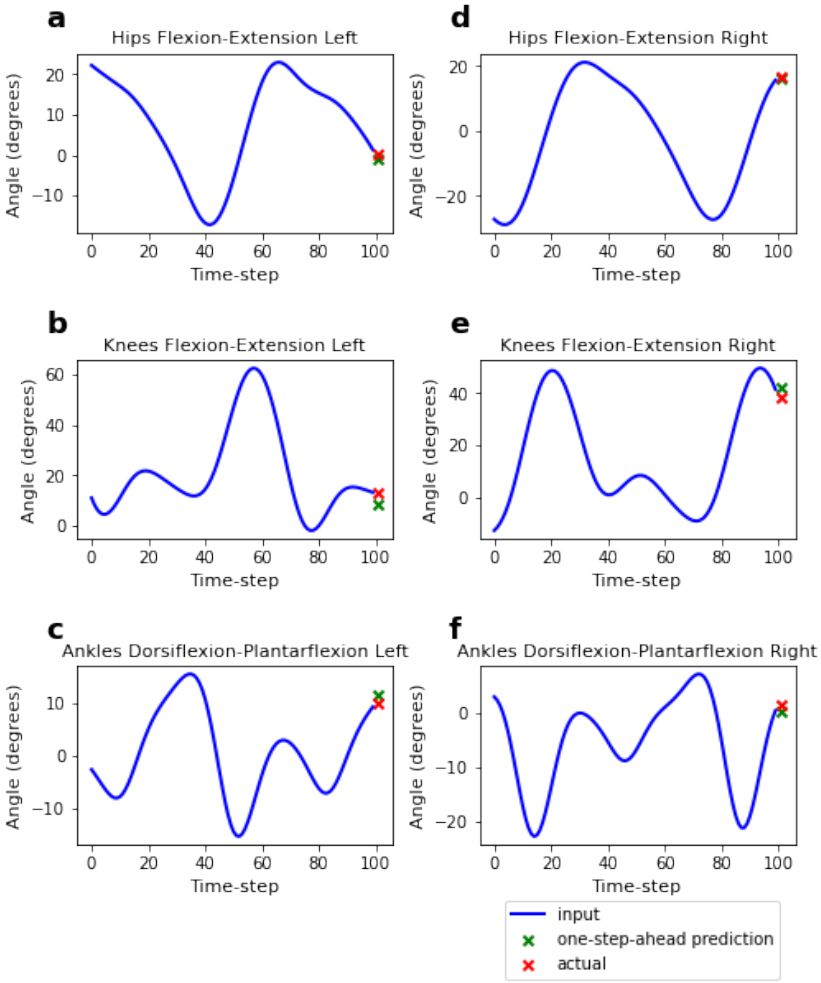

Figure 5.10: One-step-ahead prediction of the hip, knee and ankle flexion-extension angles based on a 100 time-step input window. The blue line corresponds to the model's input, the green and red markers correspond to the predicted and actual gait values, respectively. The figure shows TD gait.

According to the results, the LSTM model achieved the lowest MAE (0.87°) for one-step-ahead prediction of TD gait. The Transformer model performed slightly worse with an MAE of 1.17°, followed by the FCN model with an MAE of 1.63°. The CNN model showed the poorest performance with an MAE of 4.05°. The MAEs of the predicted values for children with CP were higher compared to TD children.

5.5.2 Performance on long-term recursive predictions

The one-step-ahead prediction models trained on typically developing children are used for long-term forecasting, by recursively using the one-step-ahead predictions as input (see Figure 5.13), to predict a total of 200 time-steps in the future. The results for

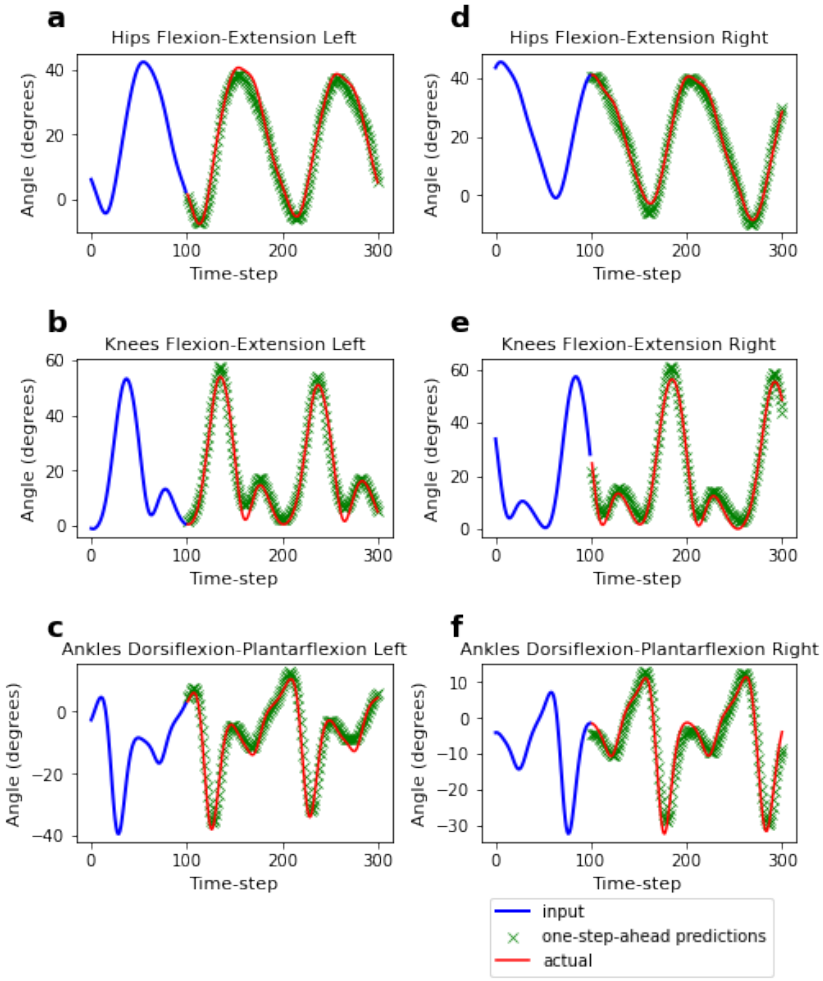

Figure 5.11: One-step-ahead prediction of the hip, knee and ankle flexion-extension angles for 200 time-steps. The blue line corresponds to the model's input, the red line to the actual gait values, and the green markers correspond to the predicted values. Each prediction (green marker) is based on a 100 time-step window of the actual/measured gait values (i.e. without recursive input). The figure shows TD gait.

Table 5.2: MAEs and MSEs for one-step-ahead gait trajectory predictions for typically developing children and children with Cerebral Palsy (in degrees)

		LSTM	FCN	CNN	Transformer
TD	MSE	1.29	4.35	43.55	2.23
	MSE std	1.92	6.17	68.14	3.25
	MAE	0.87	1.63	4.96	1.17
	MAE std	0.69	1.23	4.05	0.88
CP	MSE	3.11	11.07	88.67	5.89
	MSE std	5.74	19.46	148.23	9.82
	MAE	1.32	2.48	6.94	1.83
	MAE std	1.09	2.04	5.79	1.50

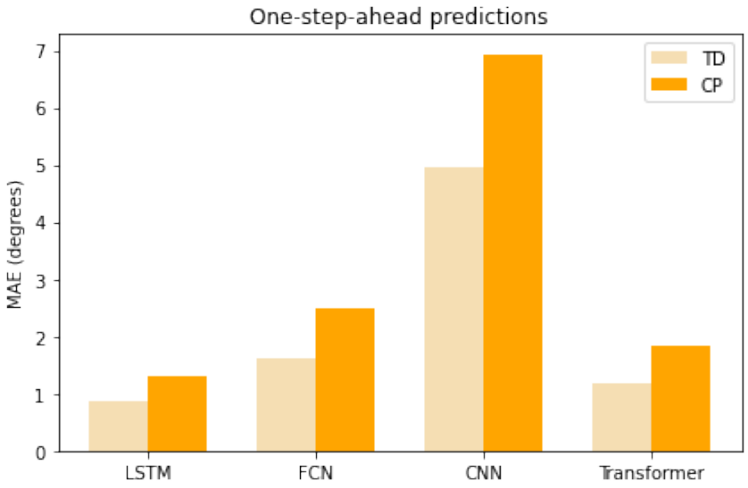

Figure 5.12: MAEs for one-step-ahead gait trajectory predictions for typically developing children and children with Cerebral Palsy (in degrees)

long-term predictions are reported in Table 5.3. For long-term predictions of TD gait, the LSTM model achieved the lowest MAE (9.36°) among the four models. The FCN model followed closely with an MAE of 10.03°, followed by the Transformer model with an MAE of 10.31°. The CNN model showed slightly higher errors with an MAE of 10.72°. For long-term predictions, the models demonstrated similar performance, with differences in errors between the different models smaller in long-term predictions than the differences in errors between the models observed in short-term predictions.

Table 5.3: MAEs and MSEs for long-term (200 time-step) recursive predictions for typically developing children and children with Cerebral Palsy (in degrees)

		LSTM	FCN	CNN	Transformer
TD	MSE	171.76	183.93	224.25	204.35
	MSE std	328.28	336.98	399.06	377.30
	MAE	9.36	10.03	10.72	10.31
	MAE std	8.62	8.90	9.64	9.29
CP	MSE	322.13	358.69	351.25	405.60
	MSE std	516.78	552.19	536.44	628.04
	MAE	13.41	14.29	14.03	14.93
	MAE std	11.32	11.72	11.55	12.51

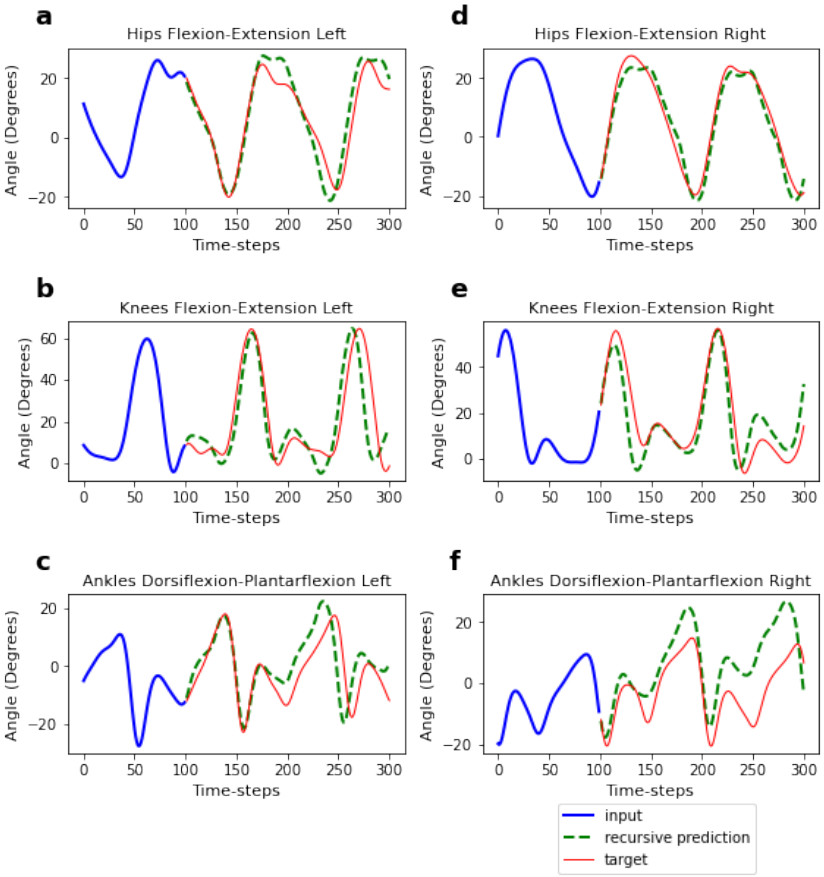

Figure 5.13: Long-term (200 time-step) recursive predictions of the hip, knee, and ankle flexion-extension angles based on a 100 time-step input window. The blue line corresponds to the model's input, the green line to the recursive predictions, and the red line to the actual gait values. The figure shows TD gait.

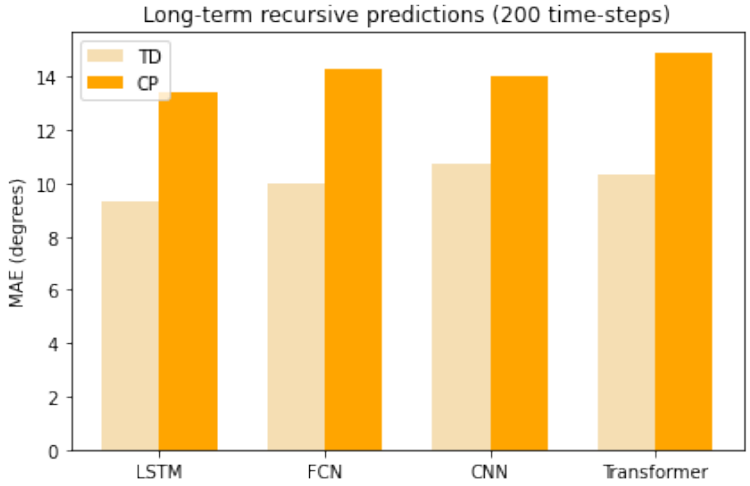

Figure 5.14: MAEs for long-term (200 time-step) recursive predictions for typically developing children and children with Cerebral Palsy (in degrees)

5.5.3 Effect of Gaussian noise on the stability of the models

Recursive predictions, generated by using on-step-ahead predictions as inputs to the models, are a way to assess the stability of the networks since the predictions will contain a level of error which will be continuously propagated. We have evaluated the performance of recursively predicting 200 time-steps in the future based on a 100 time-step input (see section 5.5.2 for details). To further investigate the effect of noise, Gaussian noise was added to the predictions (between 1-5% of the predicted value) before using it as a recursive input. The errors for long-term (200 time-step) predictions were calculated afterwards. Figure 5.15 illustrates the impact of noise on the prediction of hip flexion-extension angles. Note that noise has been added to all joint angles, and its effect is evaluated in TD and CP gait predictions.

Figure 5.16 presents the effect of Gaussian noise (levels 1-5%) on the MAEs for long-term prediction of TD gait. The results demonstrate that the errors increase linearly with increasing noise levels. Among the models, the LSTM is the most affected by noise. The FCN and Transformer are slightly more affected by noise compared to the CNN, but they are still more stable than the LSTM overall. A similar trend is observed for CP gait predictions with noise, as shown in Figure 5.17.

5.5.4 Generating adaptive reference trajectories for Cerebral Palsy Gait

This study proposed an approach that suggests corrections to cerebral palsy (CP) gait trajectories, which can be used as target/reference trajectories for position-controlled rehabilitative exoskeletons. The deep learning models were trained on typically developing (TD) gait data only and their performance for long-term predictions (200 time-steps) when CP gait was used as input was evaluated. The results reported in Figure 5.18 indicate that the models appear to introduce TD patterns onto the predicted CP gaits. Preliminary observations show that the predicted trajectories are ahead of the actual CP trajectories which indicates that the models may be imposing a higher gait speed. Figure 5.18(a) and Figure 5.18(b) illustrate this effect by showing

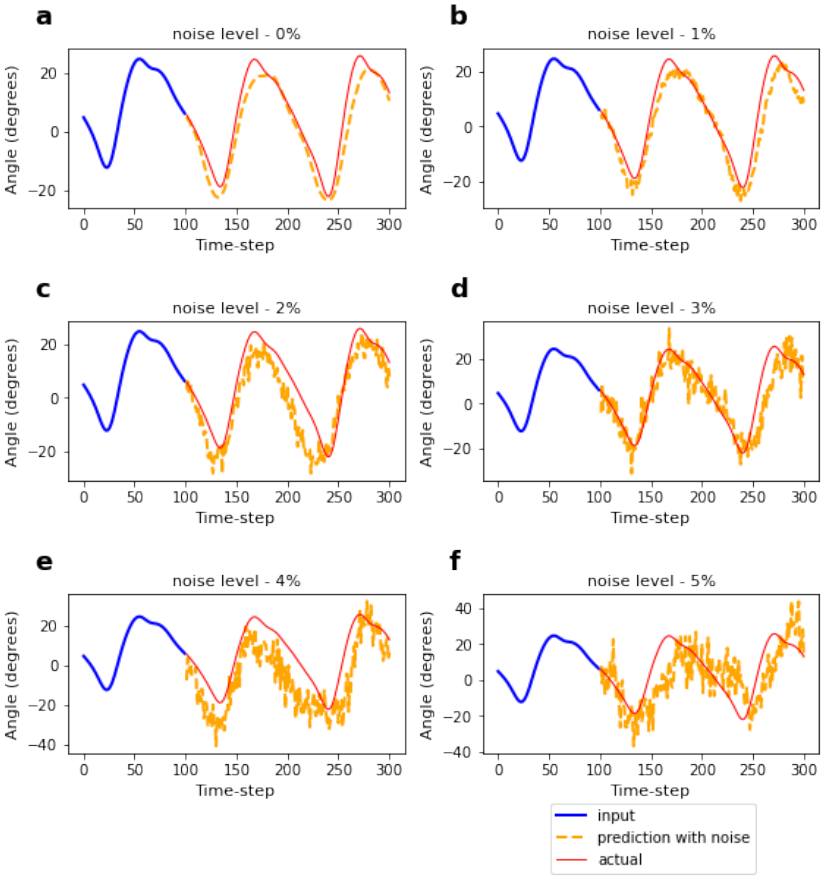

Figure 5.15: Effect of varying Gaussian noise levels (1%-5%) on the long-term recursive prediction of the hip flexion-extension angle. The blue line represents the gait input, the red line represents the actual values, and the orange line represents the predictions with added noise.

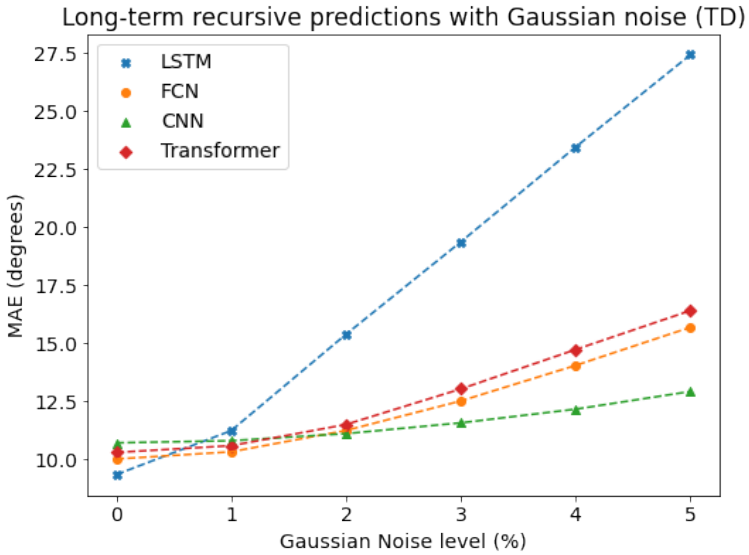

Figure 5.16: Effect of varying Gaussian noise levels (1%-5%) on MAEs for TD gait predictions

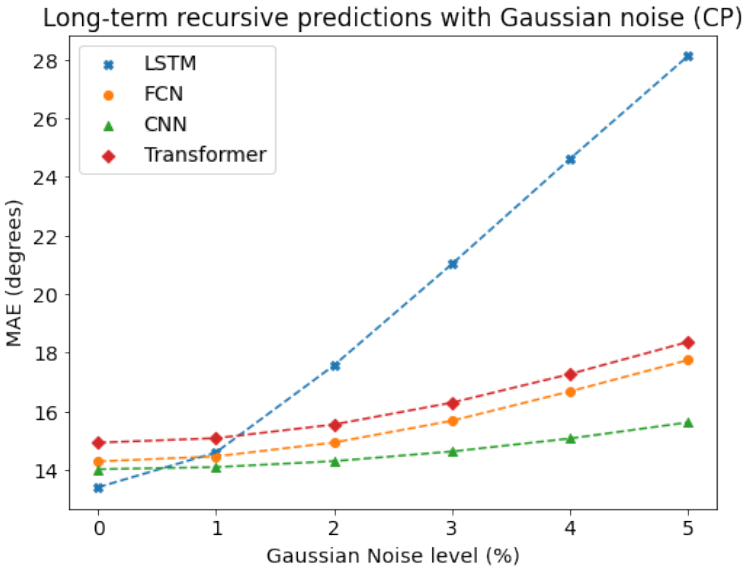

Figure 5.17: Effect of varying Gaussian noise levels (1%-5%) on MAEs for CP gait predictions

a decrease in the stride time in the predicted corrections compared to the CP gait trajectory without intervention, measured by a shorter peak-to-peak distance. The stride time in Figure 5.18(a) reduces from 137 time-steps in the CP gait trajectory without intervention to 85 time-steps in the predicted correction. Similarly, the stride time in Figure 5.18(b) was also reduced by 57 time-steps in the corrected intervention.

Furthermore, the predicted corrections indicate an increased range of motion, such as increased knee flexion, making them more similar to TD gait. This is illustrated in Figures 5.18(c), 5.18(d), 5.18(e) where the range of motion of the joint angles increased by 28.4°, 13.97°, and 19.32° respectively, in the predicted corrections compared to the CP gait without intervention.

These observations align with desired outcomes in CP rehabilitation which include increased mean velocity and improvement in knee extension [9, 13, 40]. However, it's important to note that these results are preliminary observations, and further evaluation is needed to assess the effectiveness of the generated trajectories in enhancing rehabilitation outcomes, such as reducing metabolic cost and increasing gait speed, as well as considering the comfort of users. Clinical studies in a controlled setting would be necessary to evaluate these aspects.

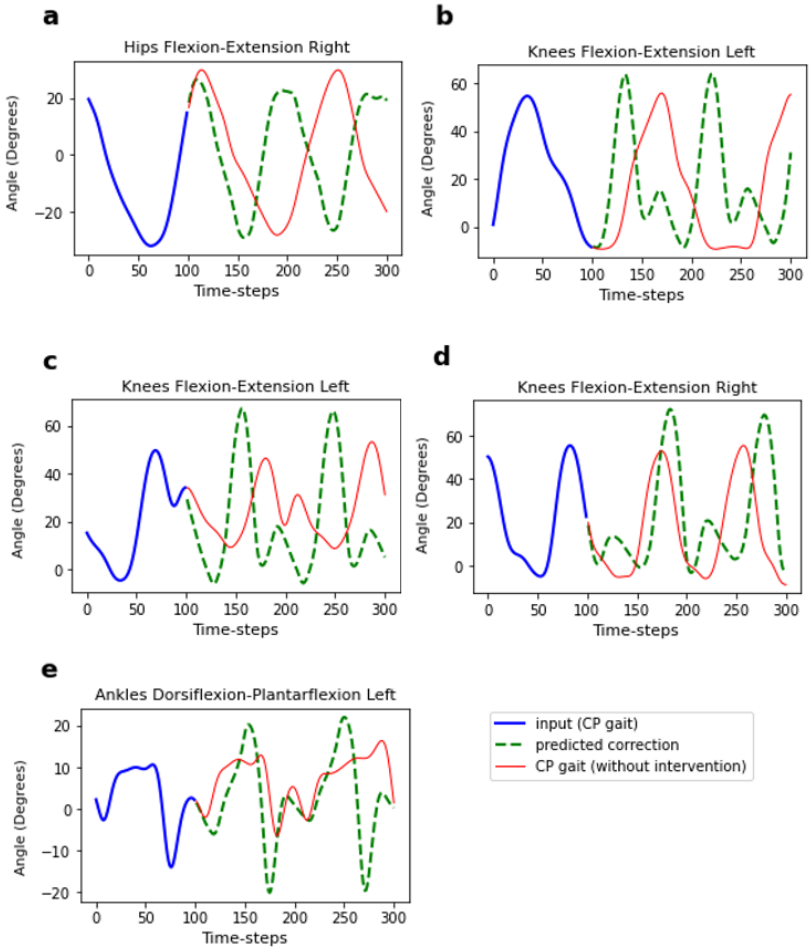

Figure 5.18: Examples of corrections to Cerebral Palsy gait predicted by a model trained on the gait patterns of typically developing children only. The blue line represents the CP gait input, the red line represents the actual CP values, and the green line represents the predicted corrections to CP gait. (a) and (b) show a decrease in the peak-to-peak distance in predicted correction compared to CP gait without intervention, suggesting that the models are imposing higher speeds. (c), (d), and (e) show an increase in the range of angles in predicted corrections compared to CP gait without intervention indicating that the models are imposing a larger range of motion.

5.6 Discussion

In this study, the focus was on developing end-to-end deep learning models for the task of gait trajectory prediction, specifically for flexion-extension angles of the hip, knee, and ankles of both the right and left legs. The intended application of these models is the control of rehabilitative exoskeletons for children with Cerebral Palsy (CP). Four deep learning models (LSTM, FCN, CNN, and Transformer) were trained for one-step-ahead predictions using a 100 time-step input window. To the best of our knowledge, this is the first time Transformers have been evaluated for gait trajectory forecasting. These models have been trained on the gait patterns of typically developing (TD) children. We proposed a methodology that optimises for long-term stability during training. This methodology involves using dynamic time warping (DTW) distances between long-term recursive predictions and true values as an early stopping metric (described in section 5.4.5.2). This has prevented the models from over-fitting on one-step-ahead predictions, at the cost of long-term stability.

The performance of these models was evaluated in terms of mean absolute errors (MAEs) for both one-step-ahead and long-term predictions. For one-step-ahead predictions of typically developing (TD) gait patterns, the MAEs ranged from 0.87° to 4.96° across all models, with LSTM achieving the lowest errors, followed by the Transformer, FCN, and CNN. The performance gap between the CNN and the other models was relatively large. For long-term recursive predictions (200 future time-steps), the MAEs on the TD gait test set ranged from 13.41° to 14.93°. The differences in performance across all models were narrower in long-term predictions, but LSTM still had the lowest errors while CNN had the largest errors.

Comparisons with prior studies in the literature are challenging due to the novelty of applying AI trajectory prediction models to pediatric gait and the gait of children with CP. Furthermore, previous studies have focused on different parameters, such as predicting linear acceleration, angular velocity, or joint moments instead of joint angles, or used different data modalities, such as EMG or a motion capture system. However, some general comparisons can still be made. For example, CNN's performance in this

study aligns with previous findings of it being less robust in gait prediction compared to other networks. These results are shown in chapter 4 [153]. But these findings were different to what was reported by Moreira *et al.* [145] who performed ankle joint torque estimation based on kinematics, speed, and anthropometry, and found the CNN to be more robust. They didn't however compare the performance to FCN or transformers. On the other hand, the LSTM's superiority over the FCN in one-step-ahead predictions is consistent with findings from other studies on joint moment prediction such as the study by Molinaro *et al.* [146] who found that the LSTM outperforms the FCN in joint moment prediction.

The study conducted has shed light on the stability of deep learning models. The impact of Gaussian noise on the models (between 1%-5% of the predicted value) was examined, revealing a linear increase in MAEs. The LSTM which had the lowest short-term and long-term errors was the most affected, while the CNN, which had the largest short-term and long-term errors, exhibited greater stability. The Transformer and FCN showed an intermediate response to noise; they were impacted slightly more than the CNN network, but less significantly compared to the LSTM network. These findings highlight the importance of minimizing noise in the system to improve prediction accuracy. This should be considered during the design of the exoskeletons.

Considering the overall results, the Transformer and FCN models emerged as more suitable options for trajectory prediction in exoskeleton control. They exhibited low errors in both short-term and long-term predictions while demonstrating better stability in the presence of noise compared to the LSTM and CNN models.

The study also proposed an approach for generating adaptive target/reference trajectories for children with CP by utilising trajectory forecasting models trained on typically developing (TD) gait data. We hypothesised that a trajectory forecasting model trained on the gait of typically developing children only, will learn their representations, and introduce corrections to CP gait patterns when used as input. Preliminary observations indicated that these models introduced TD patterns into CP gait, which aligns with desired CP rehabilitation outcomes. However, further evaluation is needed to assess the effectiveness of these reference trajectories on rehabilitation outcomes and user

comfort.

Several limitations of the study were acknowledged, including the relatively small dataset size and the need for a larger sample of TD children with a wider anthropometric distribution. The dataset used consists of 21 children (10 TD and 11 CP). We believe that a larger sample of TD children, with a wider anthropometric distribution, is needed to train the model; ideally, the anthropometrics of children with CP and of TD children used in this study should have had a more similar distribution, with a larger sample and more gait variability. Furthermore, the models were trained on flexion-extension angles obtained from IMU sensors rather than encoders which are typically used in exoskeletons.

5.7 Conclusion

The main objective of this study was to develop and evaluate deep learning algorithms for predicting gait trajectories that can be used to control exoskeletons for the rehabilitation of children with Cerebral Palsy (CP). Four deep learning models were trained on gait data from typically developing children, specifically focusing on the task of one-step-ahead prediction of gait trajectories. The goal was to leverage these predictive models to assist in the control of exoskeletons during CP rehabilitation. A methodology has been proposed to enhance the stability and accuracy of long-term forecasts, and the performance of the models was assessed in the presence of noise. The results indicated that the Transformer and Fully Connected Network (FCN) models exhibited better stability and achieved lower prediction errors, making them potentially well-suited for the intended application. Additionally, an approach for generating adaptive reference or target trajectories for position-controlled exoskeletons was introduced. The idea was to utilise the learned representations of gait patterns from typically developing children to "correct" the gait patterns observed in children with CP. Preliminary findings showed promising results, indicating that the models were able to introduce typical gait patterns to CP gaits. However, further investigations are necessary to evaluate the effectiveness of using these adaptive patterns as reference trajectories in terms of rehabilitation outcomes and user comfort. Overall, this study contributes to the

development of deep learning algorithms for gait trajectory prediction and proposes potential applications in exoskeleton control for CP rehabilitation. The models and methodologies presented have the potential to improve the control and effectiveness of exoskeletons in assisting children with CP. Future studies should focus on evaluating the practical effectiveness and impact of these approaches in clinical settings.

Chapter 6: Investigating the Effect of Gait Speed on Trajectory Prediction

6.1 Overview

In this chapter, the robustness of deep learning models, specifically fully connected neural networks, is investigated. This study involves predicting joint kinematics at varying gait speeds. The predictive performance of the fully connected network is evaluated in short-term and long-term prediction at gait speeds included in the training dataset speed range, and on speeds excluded from it.

6.2 Introduction

AI models require training on datasets and in the case of exoskeleton applications, the training dataset could include gait parameters such as joint kinetics, joint kinematics, foot pressure, and muscle activity [19]. AI models utilise these parameters as input to make predictions or classifications, depending on the task. It is crucial for the training dataset to be a representative sample of the data that the exoskeleton is expected to receive as input during real-life operation. Depending on the data used for training and testing the model, we can develop individualised (dependent), generalised (independent), and semi-dependent models [74]. A dependent model needs to be trained on data from an exoskeleton user before using the exoskeleton. An independent model, on the other hand, does not require training on data from each exoskeleton user but

is instead trained on data from multiple individuals. Semi-dependent models combine both approaches, where a model is trained on data from multiple individuals but fine-tuned using data from the specific user before using the exoskeleton [74]. Although individualised models have demonstrated higher accuracy in performance compared to generalised models [74], it is not always possible or practical to fine-tune models for each user, which is why generalised models are sometimes adopted.

It is essential to consider how the models, often trained on data from able-bodied users, would perform when controlling assistive and rehabilitative exoskeletons for users with pathological gaits. The target users of these exoskeletons typically walk at slower gait speeds [165]. For instance, able-bodied adults have walking gait speeds between 0.75 and 1.75 m/s, whereas adult stroke patients walk at much lower speeds, ranging between 0.08 and 1.05 m/s [166]. Additionally, over the course of a rehabilitation session, a patient's speed may change, with their mean velocity increasing [13].

The main contribution of this study is to investigate the effect of speed on the prediction of gait trajectories for able-bodied users, as well as the performance of the models in predicting trajectories at gait speeds that are excluded from the training speed range (see Section 6.4 for details). The performance of the models is compared on gait speeds included in the training range, as well as speeds excluded from it. This investigation enables us to assess the generalisability of the models when tested on gaits from users walking at speeds that have not been included in the training dataset and to examine the influence of speed on gait. The findings of this study can serve as guidance for developers of exoskeletons, informing their decision on which speeds to include when collecting data for training AI models for exoskeleton control.

6.3 Background

Gait speed is known to impact spatiotemporal parameters (cadence, step length, and stride length), joint kinetics, ground reaction forces, and joint kinematics [167]. The magnitude of this impact varies among children, young adults, and older adults [167]. The effect of gait speed on joint kinematics has been shown to be moderate to large

[167]. In young adults, for example, gait speed has been shown to impact the minimum and maximum joint angle values, specifically increasing hip flexion, hip extension, knee flexion, and ankle plantar-flexion angles with higher speeds [167]. Furthermore, gait speed appears to have a more significant impact on the kinematics of children, whose gait patterns have not fully matured, compared to individuals from other age groups [167, 168].

Since speed is a vital biomechanical determinant of gait patterns [165], it is essential to consider it when generating reference gait patterns for position-controlled exoskeletons. Fukuchi et al. [165] utilised regression to generate normalised reference gait patterns that were speed-dependent. They aimed to establish a database of reference gait patterns at varying speeds that could be used to assess the gait patterns of individuals with gait pathologies, who often walk at lower speeds compared to able-bodied individuals [169, 170]. Zaroug et al. [136] assessed the performance of deep learning models in predicting lower limb kinematics at speeds 20% lower and higher than preferred walking speeds. The performance of their models decreased when predicting gait at slower speeds but increased when predicting gait at faster speeds. Apart from speed, anthropometric parameters can also influence gait patterns [29, 159, 171]. Zou et al. [172] developed a two-step method for gait trajectory prediction based on an individual's unique anthropometrics and desired speed during rehabilitation. Their approach consisted of a Gait Parameter Model (GPM), which is a neural network that selects gait parameters based on anthropometrics and speed, and a Gait Trajectory Model (GTM), which uses these parameters, along with kernelised movement primitives, to reconstruct the reference gait patterns for an exoskeleton. Han et al. [173] implemented Future Generative Adversarial Nets (F-SeqGAN) trained on varying gait speeds for gait trajectory prediction, even during acceleration, without the need to pre-define the input speed. Embry et al. [174] developed a basic model that can continuously predict joint kinematics based on gait phase, speed, and inclination.

Although many of the existing approaches test their models on gait speeds included in the training speed range, only a few have explored how the models perform and extrapolate to speeds that are excluded from the training set. For the task of gait phase prediction, Lu et al. [175] observed a decrease in the performance of a long short-

term memory (LSTM) network on speeds not included in the training set. However, their study was limited to only two subjects, and the training data were biased towards constant speeds. Meanwhile, Kang et al. [74] implemented neural networks for gait phase prediction and found that their models were capable of extrapolating to higher speeds, with the semi-dependent model outperforming the dependent and independent models. Their results were based on 10 subjects, but they only extrapolated to speeds 0.1 m/s higher than the training speed range. The two aforementioned studies assessed the ability of the models to classify gait phases on speeds excluded from the training speed range, but either had a low number of participants or extrapolated to speeds that were only slightly beyond the training speed range. Furthermore, no studies have investigated the task of gait trajectory prediction, where the performance of models is evaluated for gait speeds that are both lower and higher than the training speed range.

To address these gaps, this study aims to evaluate and compare the performance of deep learning models when tested on speeds that are both included and excluded from the training speed range. This assessment may help determine the transferability of the models to real-world applications, where an exoskeleton may need to operate in an environment with greater variability than what it was initially trained for.

6.4 Methodology

6.4.1 Overview

This study evaluates how fully connected neural networks (FCNs) for gait trajectory prediction extrapolate to speeds excluded from the training range. The FCNs were trained to perform one-step-ahead predictions of joint kinematics, focusing on the angles of the hip, knee, and ankle for both the left and right legs. Predictions were based on a short input window of past joint kinematic values. Four FCNs were trained and developed using data obtained at varying gait speeds. Subsequently, the performance of the models was evaluated by testing them on gait speeds included and excluded from the training speed range (refer to Table 6.1 for further details).

Table 6.1: Range of speeds FCNs were trained and tested on (low speeds 0.5 m/s - 1.0 m/s, medium speeds 1.05 m/s - 1.45 m/s, and high speeds 1.5 m/s - 1.85 m/s)

Model	Training Set Speed Ranges (m/s) (included speeds)	Testing Set Speed Ranges (m/s) (excluded speeds)
Generalised speed model	all (0.5 - 1.85)	all (0.5 - 1.85)
Low-speed model	low and medium (0.5 - 1.45)	high (1.50 - 1.85)
High-speed model	high and medium (1.05 - 1.85)	low (0.5 - 1.0)
Low-high speed model	low and high (0.5 - 1.0, 1.50 - 1.85)	medium (1.05 - 1.45)

6.4.2 Data

The FCNs in this study were trained using an online gait dataset by Camargo et al. [128]. Details on this dataset can be found in Chapter 3, section 3.2.3.

6.4.3 Pre-processing

In order to evaluate the predictive performance of the FCNs at speeds excluded from the training range, the data is segmented into three distinct speed ranges: low speeds ranging from 0.5 m/s to 1.0 m/s, medium speeds ranging from 1.05 m/s to 1.45 m/s, and high speeds ranging from 1.5 m/s to 1.85 m/s. These speed ranges were determined based on the low-, comfortable-, and high-speed ranges reported in the literature for young adults [167, 176, 177].

Four FCNs were developed and evaluated in this study. They had the same architecture (including the number of layers and nodes per layer) but each was trained on gait data at varying speeds. The respective training and testing speed ranges for each of the models were as follows: (1) the *generalised-speed model* was trained and tested on all gait speed ranges, (2) the *high-speed model* was trained on medium and high speeds and tested on low speeds, (3) the *low-speed model* was trained on low and medium speeds and tested on high speeds, and (4) the *low-high-speed model* was trained on low and high speeds and tested on medium speeds. Details about the FCN models and their corresponding training and testing speed ranges can be found in Table 6.1.

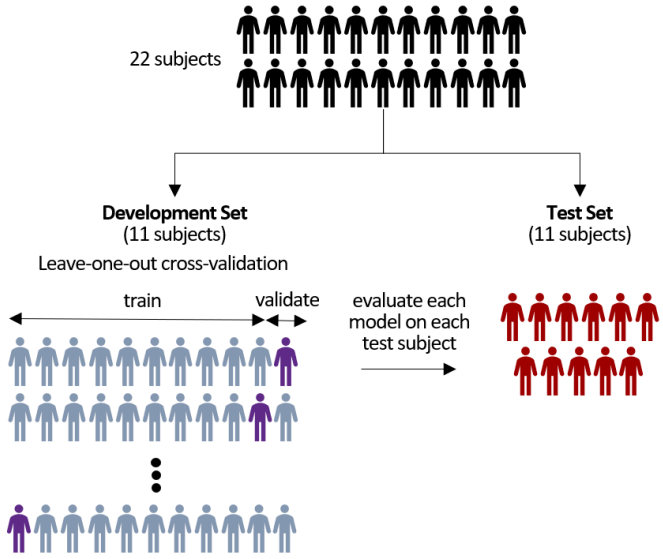

Figure 6.1: Illustration of the train-test split used for training and evaluating the FCNs, using the leave-one-out cross-validation method.

The data from the 22 able-bodied individuals were randomly split into two sets: a development set and a test set. This division was performed at the subject level, with 11 subjects in each set. The development set was used for two main purposes: hyperparameter optimisation and model training. For hyperparameter optimisation, 70% of the subjects in the development set were used for training (8 subjects), whereas the remaining 30% (3 subjects) were used for validation. After the hyperparameter optimisation phase, cross-validation was performed using the leave-one-subject-out method. In this process, the FCNs were trained on data from 10 subjects in the development set, with the data from the 11th subject left out for validation. This procedure was repeated 11 times, each time using a different subject as the validation subject. The test set, consisting of data from unseen subjects, was used to evaluate the performance of the models each time (see Figure 6.1). This evaluation included assessing the performance of the models on the gait speeds included in the training range, as well as on outlier speeds.

The windowing method, introduced in chapter 4, was used to generate training and testing samples for each set in the study [153]. The stride length, which determines the number of training samples that will be derived from each gait sequence, was varied for the different FCN models to ensure that all models, including the generalised-, low-, high-, and low-high-speed models, were trained on a similar number of samples. Each

training sample consisted of an input matrix, x_{in}, and target vector y_{out}. x_{in} represents a 200-time-step window of joint angle values, including the hip, knee, and ankle angles for the left and right legs in the sagittal plane. The input window corresponds to 1 s of data for a sampling frequency of 200Hz. y_{out} represents the values of the joint angles (hip, knee, and ankle angles for the left and right feet) for the next time-step. The output window size is 1 time-step, representing the immediate future joint angle values that the FCN models are trained to predict.

For n samples in a set, $X_{in} \in \mathbb{R}^{n \times l_{in} \times f}$, l_{in} (set to 200) is the input window size and f (set to 6) is the number of input features (kinematic joint angles). Similarly, in $Y_{out} \in \mathbb{R}^{n \times l_{out} \times f}$, l_{out} (set to 1) is the output window size and f (set to 6) is the number of output features.

The FCN models (generalised-speed, low-speed, high-speed, and low-high-speed models) were trained and tested on different gait speeds. All inputs to the models and the corresponding target outputs were normalised using min-max normalisation such that $X_{in} \in [0, 1]$ and $Y_{out} \in [0, 1]$. The min-max values were chosen to accommodate for the minimum and maximum values of all the individuals in the dataset, with an additional safety boundary to ensure the data fell within the normalised range.

6.4.4 Model Architecture and Optimisation

According to the findings in chapter 5, the fully connected neural network (FCN) demonstrated low errors in both short-term and long-term gait trajectory prediction tasks and exhibited higher robustness to added noise [178]. These were the reasons for selecting the FCN for this study. The input to the FCN, which was 2-dimensional $\mathbb{R}^{200 \times 6}$ as it included the values of 6 joint angles for a 200-time-step window, was flattened into a 1-dimensional vector $\mathbb{R}^{1200}$ and passed through 5 fully connected linear layers with ReLU activation functions in between. The FCN architecture is illustrated in Figure 6.2. A note on how the FCN in this chapter differs from the ones implemented in Chapters 4 and 5 is included in Chapter 4, section 4.3.5.

For training, each of the models was optimised to minimise the mean squared error

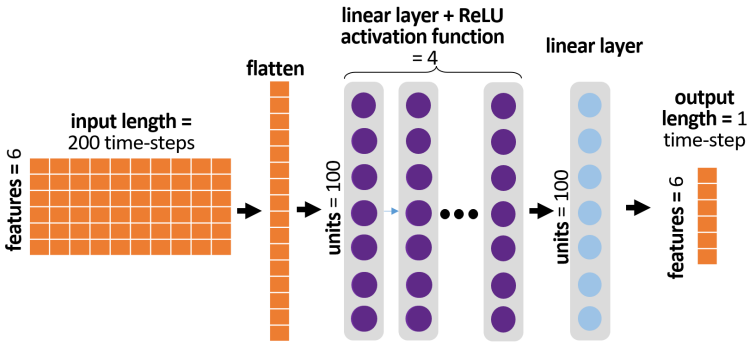

Figure 6.2: Fully Connected Neural Network (FCN) Architecture

(MSE) difference between the predictions and the target one-step-ahead kinematic values using the Adam optimiser. The optimal hyperparameters, including the learning rate, number of layers, number of nodes per layer, and batch size, were selected based on the tree-structured Parzen estimator algorithm, a type of Bayesian hyperparameter sampler. Hyperparameter optimisation (HPO) was applied to the generalised model, and the optimised architecture resulting from this process was used for the other models (low-speed, high-speed, and low-high-speed models). The search space and corresponding selected values are shown in Table 6.2.

During training, the dynamic time-warping (DTW) distances were calculated between 200 recursively predicted time-steps and the true gait values after each training epoch. The DTW distances were used as a metric to determine when to end the training of the models. The models were trained for 70 epochs, with training being terminated earlier (using the early stopping method) if the DTW distances on the validation set did not decrease for 20 epochs (refer to chapter 5 which elaborates on how DTW distances were used to optimise gait trajectory prediction models [178]).

The Pytorch machine learning framework was used in this study, along with various libraries, including Matplotlib, Numpy, Seaborn, SciPy, Scikit-Posthocs, and Optuna, for hyperparameter optimisation [142]. The DTW Python package was used for calculating dynamic time-warping distances [164]. The computations were performed using an Nvidia Geforce RTX 2070 GPU.

Table 6.2: Search space for FCN hyper-parameters and the selected values

Hyper-parameter	Search Space	Selected Value
learning rate	[0.01, 0.001, 0.0001, 0.00001]	0.00001
number of layers	[3, 4, 5, 8, 10, 12]	5
nodes per layer	[10, 30, 70, 100, 150, 200]	100
batch size	[32, 64, 128, 256, 512]	32

6.4.5 Evaluation Metrics and Statistical Analysis

The mean absolute error (MAE) and mean squared error (MSE) were used as the evaluation metrics to assess the performance of the models. These metrics were calculated for both short-term predictions, comparing one predicted time-step to the actual gait values, and long-term predictions, comparing 200 time-steps of gait values generated using recursive forecasting (where predictions are fed back as input to generate further predictions) to the actual gait values. The MAEs and MSEs were calculated after the de-normalisation of the models' outputs. The MAEs were calculated using equation 4.7, while MSEs were calculated using equation 4.9, both introduced in Chapter 4, section 4.3.7.

Section 6.5 reports the results of the statistical tests conducted. The tests were carried out in Python using the Scipy and Scikit-Posthocs libraries.

6.5 Results

Firstly, the performance of the four different models (generalised-speed, low-speed, high-speed, and low-high-speed models) was evaluated on speeds included in the training range but from unseen subjects. The MAEs and MSEs for the short-term (1-time-step) and long-term (200-time-step) predictions are presented in Table 6.3. For the short-term predictions (depicted in Figure 6.3), the MAEs ranged from 1.21° to 1.34°, with a maximum MAE difference of 0.14° across the four models. The low-speed model showed slightly higher errors compared to the other three models (see Figure 6.4a,

Table 6.3: MSE and MAE for 1-step-ahead and 200 time-step gait trajectory predictions for generalised, low, high, and low-high speed models evaluated on speeds included in the training range (in degrees).

		Generalised speed model	Low speed model	High speed model	Low-high speed model
1-time-step	MSE	2.58	2.66	2.62	3.43
	MSE std	0.99	1.08	1.32	2.58
	MAE	1.21	1.22	1.21	1.34
	MAE std	0.24	0.25	0.30	0.46
200-time-steps	MSE	60.84	58.36	43.13	63.25
	MSE std	25.97	22.00	24.18	23.92
	MAE	5.24	5.13	4.42	5.39
	MAE std	1.04	0.98	1.06	0.97

illustrating the MAE difference for short-term predictions). Meanwhile, for the long-term predictions (depicted in Figure 6.5), the MAEs were higher, ranging from 4.42° to 5.39°, with a maximum difference of 0.97° across the four models. The low-high-speed model exhibited the highest MAEs, whereas the high-speed model exhibited the lowest MAEs (see Figure 6.4b, illustrating the MAE difference for long-term predictions). All differences in the one-step-ahead prediction errors on speeds included in the training range among the various FCNs were statistically significant. The statistical significance was determined based on the Kruskal–Wallis H-test (p <0.05), followed by Dunn's post hoc test for pairwise comparisons.

The performance of the four models was subsequently evaluated on speeds excluded from the training range and from unseen subjects. The MAEs and MSEs for the short- and long-term predictions on excluded speeds are presented in Table 6.4. For the short-term predictions on excluded speeds, the MAEs ranged from 1.31° to 2.03° across the four models, and from 4.86° to 8.42° for the long-term predictions. We compared the difference in the performance of the models on speeds included in the training range and speeds excluded from the training range (see Figure 6.6). It was observed that the performance of the low- and high-speed models worsened when tested on the excluded speeds. For the one-step-ahead predictions, the MAE of the low-speed model on the excluded speeds was 66.2% higher compared to the MAE on speeds included in the training range, whereas the MAE of the high-speed model was 43.7% higher. For the long-term predictions, the MAE of the low-speed model on the excluded speeds

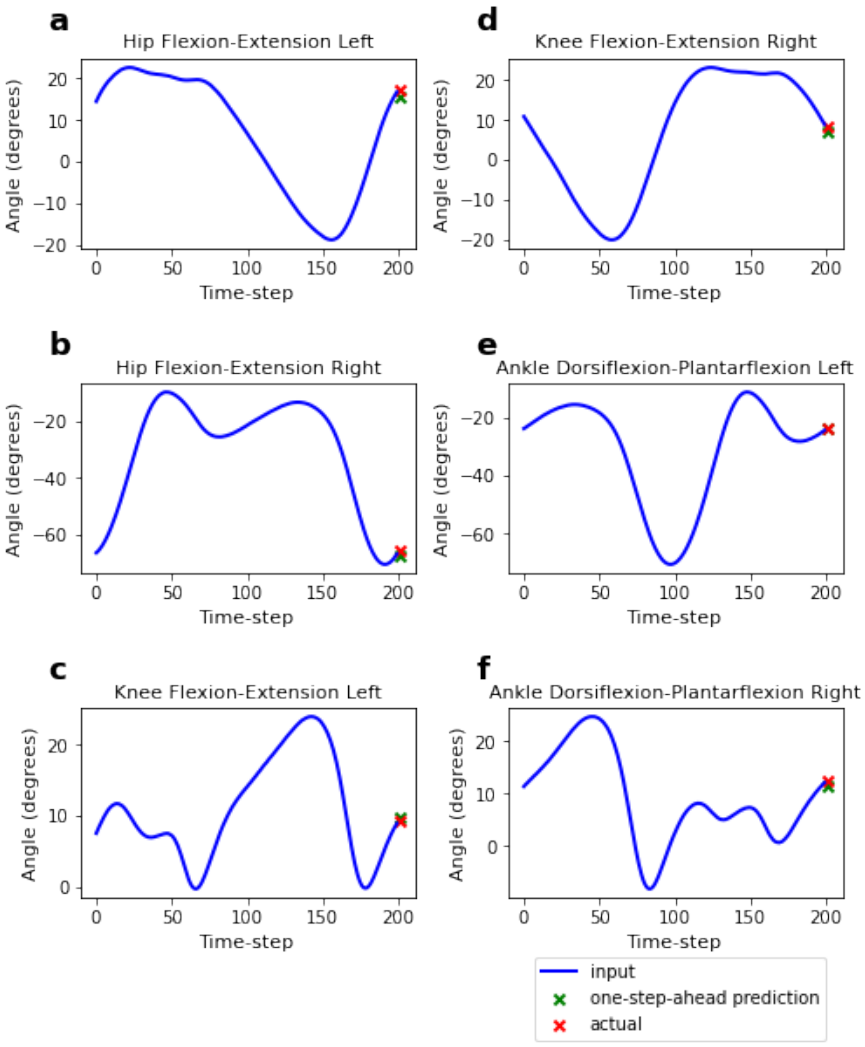

Figure 6.3: Short-term (one-step-ahead) prediction of the flexion-extension angles of the hip, knee, and ankle. Predictions (green marker) are made based on a 200 time-step input to the model (blue line), and compared to the actual values (red marker).

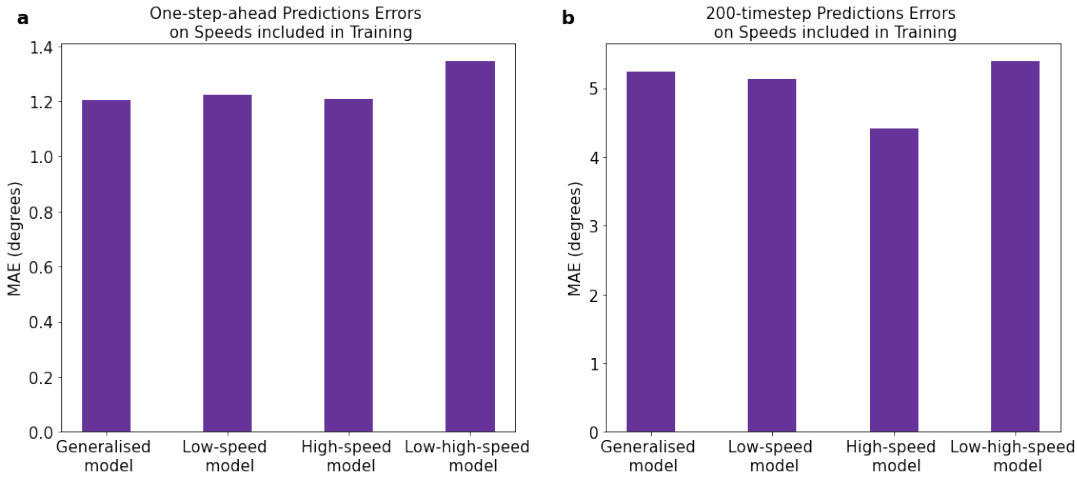

Figure 6.4: Prediction errors on speeds included in training range. (a) shows errors on short-term (1-step-ahead) predictions, while (b) shows errors on long-term (200 time-step) recursive predictions.

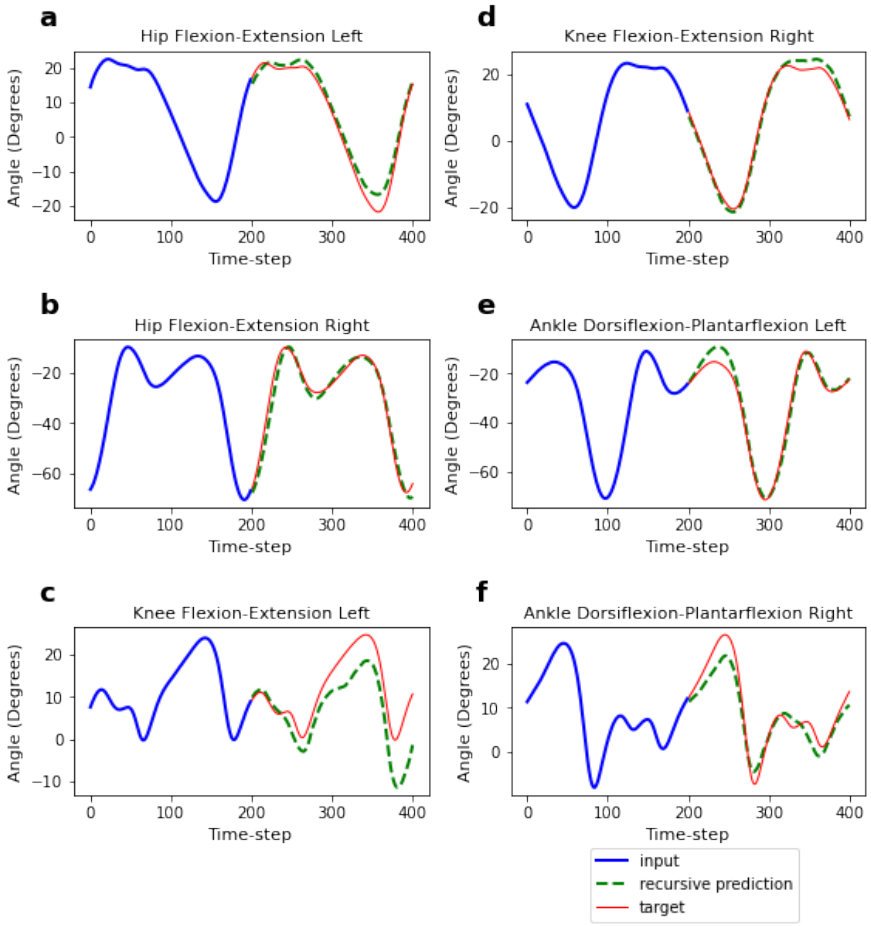

Figure 6.5: Long-term (200 time-step) prediction of the flexion-extension angles of the hip, knee, and ankle. Recursive predictions (green line) are made based on a 200 time-step input to the model (blue line), and compared to the actual values (red line).

Table 6.4: MSE and MAE for 1-step-ahead and 200 time-step gait trajectory predictions for low, high, and low-high speed models on speeds excluded from the training range (in degrees).

		Low speed model	high speed model	Low-high speed model
1-timestep	MSE	7.40	5.41	3.21
	MSE std	4.64	3.18	2.50
	MAE	2.03	1.74	1.31
	MAE std	0.63	0.54	0.44
200-timesteps	MSE	142.49	152.09	49.85
	MSE std	107.71	65.94	21.87
	MAE	7.92	8.42	4.86
	MAE std	3.06	2.02	1.06

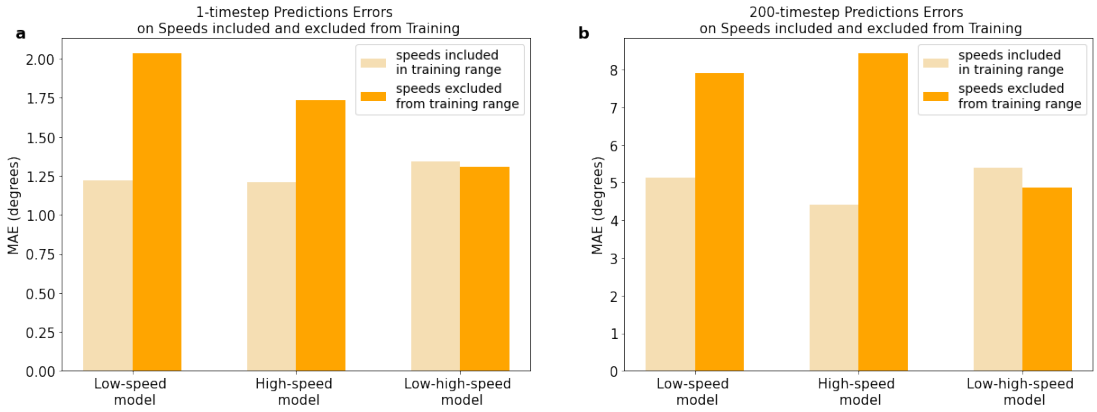

Figure 6.6: Comparison of prediction errors for gait speeds included in training range and for excluded speeds. (a) shows errors on short-term (1-step-ahead) predictions, while (b) shows errors on long-term (200 time-step) recursive predictions. All differences in prediction errors (MAEs) between included and excluded speeds for each FCN model are statistically significant for both short-term and long-term predictions (significance based on Kruskal-Wallis H-test ($p < 0.05$))

increased by 54.3% compared to the MAE on the included speeds, whereas the MAE of the high-speed model increased by 90.7%. However, the low-high-speed model showed different outcomes. It performed better on the excluded speeds (i.e., medium-speed ranges) compared to the included speeds (i.e., low- and high-speed ranges). In fact, the MAE for the excluded speeds decreased by 2.8% for the short-term predictions, compared to the MAE for the included speeds, and by 9.8% for the long-term predictions. All differences in the prediction errors (MAEs) between the included and excluded speeds for each FCN model were statistically significant for both the short- and long-term predictions. Statistical significance was determined based on the Kruskal–Wallis H-test ($p < 0.05$).

6.6 Discussion

In this study, fully-connected-neural networks (FCN) were implemented for gait trajectory prediction. The FCNs were trained on four different gait speed ranges and evaluated on speeds included and excluded from the training set. The results showed that the performance of the generalised-, low-, high-, and low-high-speed models was very similar when evaluated on the speeds they were trained on. The low-high-speed model exhibited slightly higher errors compared to the other three models for both short-term and long-term predictions. On the other hand, the high-speed model exhibited slightly lower errors, especially for long-term predictions. The findings are consistent with the results of a previous study by Zaroug et al. [136], which reported improved performance for lower limb kinematic predictions at higher gait speeds.

In comparison to related studies, Kang et al. [74] reported good performance of their gait-phase estimation model on extrapolated speeds, although they only evaluated speeds slightly higher than their training speed ranges. Meanwhile, Lu et al. [175] evaluated their continuous gait-phase recognition algorithm on untrained speeds and observed a decline in performance, but their results were based on a small number of subjects. In this study, the performance of the low- and high-speed FCN models worsened when evaluated on excluded speeds, as indicated by an increase in mean squared errors (MSEs) and mean absolute errors (MAEs) for both short-term and long-term predictions. The MAEs increased by 43.7% to 90.7% for the low- and high-speed models when compared to the MAEs on trained speeds. Interestingly, the low-high-speed model, which was trained on low- and high-speed ranges only, performed well on medium speeds. In fact, the MAEs for medium speeds improved by 2.8% for short-term predictions and 9.8% for long-term predictions compared to the low- and high-speed ranges the model was trained on. These results suggest that FCNs are capable of interpolating to speeds that lie between the maximum and minimum training speeds, even if they have not been explicitly trained on those speeds. However, they are unable to extrapolate to speeds beyond or below the maximum and minimum speeds of the training range.

One important limitation of this study is the relatively small size of the dataset used to develop and test our models, which included data from 22 subjects. This limitation is a common challenge for many applications involving human data, as data scarcity often arises due to practical constraints and limited resources. To mitigate the risk of overfitting the models and assess their generalisability across different subjects, leave-one-out cross-validation was implemented. Nonetheless, it is still necessary to validate the findings with FCNs that are trained on larger datasets. Advancements in wearable technology that enable continuous data collection may assist in addressing the issue of data scarcity. Another limitation of this study is that the data used to train the models were collected from gait cycles performed at constant speeds. To improve the implementation, it would be beneficial to include data from gait cycles with dynamic speeds, as this better reflects real-life walking conditions. Additionally, the study only considered data captured on even surfaces, without accounting for inclinations or unevenness that may be encountered in outdoor environments.

6.7 Conclusions

This study explored the performance of fully connected neural networks (FCNs) in predicting gait trajectories across different speed ranges. It examined both short-term and long-term predictions and evaluated the models on speeds that were both included and excluded from the training gait speed range. The results revealed that the FCN models exhibited a decline in performance when predicting kinematic joint trajectories at gait speeds significantly higher or lower than the training speed ranges. However, the FCN models demonstrated satisfactory performance on speeds within the maximum and minimum speed ranges, even if those speeds were not included in the training dataset. These findings highlight the importance of considering the range of speeds that an exoskeleton may encounter in real-life applications during the training and development of the models. They also emphasise the need for the development of explainable AI techniques to gain insights into the influential input features and limitations affecting the model's performance. This information can enrich our knowledge of gait analysis and biomechanics, leading to improved interventions for gait assistance and rehabilitation.

Chapter 7: Concluding Remarks and Future Directions

7.1 Overview

This chapter provides a summary of the key findings of this research. It includes concluding remarks, highlights limitations, and suggests potential future research directions.

7.2 Summary of Key Findings

This research was driven by the need to develop control algorithms for rehabilitative exoskeletons for children with Cerebral Palsy. The aim was to propose strategies to improve the control of those exoskeletons through the integration of deep-learning models. Improved control could result in enhanced human-robot interaction and has the potential to lead to increased user comfort, improved functional outcomes of rehabilitation, and reduced incidence of accidents and injuries.

This research work started with conducting a literature review (Chapter 2), where the existing use of intelligent algorithms in gait analysis for robotic control was systematically explored. Through this review, it was found that the gait parameters that are predicted or classified by AI models depend on the type of exoskeleton control, the exoskeleton's functional purpose, and the specific condition or disease of the patient.

Among the reviewed studies, the gait phase accounted for 35.6% of the predicted parameters, locomotion mode for 31.1%, joint kinematics for 24.4%, and joint kinetics (torque/moment) for 8.9%. Additionally, among the various machine learning models used, deep learning was implemented in 15% of the reviewed studies. An important gap identified in the literature is the lack of studies that included pathological gait for the development and assessment of their models. Rehabilitative and assistive exoskeletons that target individuals with pathological gait need to be tailored for and assessed on the gait of these users, who inherently have higher inter and intra-subject gait variability compared to healthy gait.

This research addresses the existing gap in the literature, by developing AI models that predict the gait trajectories of children with neurological disorders (Chapter 4). For this task, two deep learning models were implemented, a convolutional neural network and a long-short-term-memory network. For training and evaluating the models, a dataset containing gait recordings of children with neurological disorders was utilised, 73% of which had Cerebral Palsy. The influence of the size of input and output on predictions was investigated. A total of 35 combinations of input and output time-frames were examined, with window sizes for input vectors ranging from 50 to 1000 ms, and output vectors from 8.33 to 200 ms. The results showed that the input size has no significant influence on mean prediction errors when the output window is 50 ms or smaller. For output window sizes greater than 50 ms, the larger the input window, the lower the error. This suggests that for longer prediction horizons, the input window size should be increased for better performance. The study in Chapter 4 and its findings address research questions RQ1 and RQ2 presented in section 1.4.

Additionally, the performance of deep learning models for predicting the gait of typically developing children and children with Cerebral Palsy was explored (Chapter 5). The study compared the predictive capabilities of four deep learning models (Transformer, long-short-term-memory network, convolutional neural network, and fully connected neural network) in forecasting the gait trajectories of these two distinct populations, for short-term and long-term (recursive) predictions. The study introduced an approach to optimise the stability of long-term predictions. The stability of the models was subsequently assessed in the presence of varying levels of Gaussian noise. The results

revealed that the errors, measured by mean absolute error (MAE) increase linearly with higher noise levels. Based on the results of this study, the Transformer and fully connected neural network emerged as the most suitable deep learning models for trajectory predictions in the context of exoskeleton control. These models exhibited low errors in short-term (one-step-ahead) and long-term prediction tasks while being more stable in the presence of added noise. These findings address research questions RQ2 and RQ3 presented in section 1.4.

In Chapter 5, an approach for generating adaptive target/reference trajectories for children with Cerebral Palsy was also proposed. It was hypothesised that a trajectory forecasting model trained on the gait of typically developing children only, will learn their representations, and introduce corrections to CP gait patterns when used as input. Preliminary observations showed that the predicted trajectories are ahead of the actual CP trajectories, which indicates that the models may be imposing a higher gait speed. Furthermore, the predicted corrections demonstrated an increased range of motion, such as an improvement in knee flexion, making them more similar to the gait patterns of typically developing children. Those observations follow desired CP rehabilitation outcomes which include an increase in mean velocity and an improvement in knee extension and may suggest that the models are introducing TD patterns to CP gait. The results address research question RQ 4, included in section 1.4.

This research work also investigated the robustness of deep learning models in predicting kinematic trajectories at varying speeds, particularly at speeds within the training speed range and speeds outside of it (Chapter 6). The findings showed that the performance of the low and high-speed models, measured by mean absolute errors (MAE), decreased by approximately 43.7% to 90.7% when tested on speeds excluded from the training dataset. Meanwhile, the performance of the low-high-speed model (trained on low and high speeds only) improved when tested on excluded medium speeds by 2.8% for short-term predictions and by 9.8% for long-term predictions. The results suggested that the FCNs are able to interpolate to speeds that lie between the maximum and minimum training speed ranges, even if they have not been explicitly trained on those speeds. However, the predictive performance of the models decreases when predicting gait at speeds beyond or below the maximum and minimum training speed range. The

findings address research question RQ5 in section 1.4.

7.3 Concluding Remarks

To conclude, the primary aim of this research was to develop and advance AI systems designed and optimized for controlling rehabilitative exoskeletons for children with neurological disorders, specifically Cerebral Palsy. To address this objective, several research studies were conducted. AI systems, based on state-of-the-art network architectures, were developed and optimised to forecast gait trajectories using data from typically developing children and children with neurological gait. These studies explore various characteristics of AI systems, including how the predictive performance is impacted by variables such as the input and output length and the addition of noise, as well as the robustness in predicting gait at varying speed ranges, both within and outside the range of gait speeds used for training. The conclusions drawn from these studies can be applied to the development of high-level control systems for exoskeletons targeted for Cerebral Palsy gait rehabilitation. Additionally, throughout this research, an approach for generating adaptive and individualised gait patterns for children with Cerebral Palsy was proposed, with the potential to improve the control and effectiveness of rehabilitative exoskeletons. Despite these achievements, our research has some limitations, and several future research directions can be proposed based on our findings, both of which are outlined below.

7.4 Limitations

There have been some limitations in our research work, which we discuss in this section.

- Collecting gait data from children with Cerebral Palsy is a challenging task. Therefore, the models implemented in this study have been trained on data from a relatively small number of participants. Data scarcity is a common challenge in many applications involving human data, especially when dealing with patients. This limitation is often due to practical constraints and limited resources [21].

Having a larger sample size would greatly benefit the improvement of model performance given that deep learning models are data-driven. Additionally, a larger sample size would allow for a better assessment of the models' generalisability on a larger population. Cerebral palsy, in particular, manifests itself in various forms and there is a four-level gross motor function classification (GMFC) for CP, with each level corresponding to the severity of gait impairments. Furthermore, a recent Delphi consensus study has identified 49 different joint patterns in CP, including distinct patterns in the sagittal, transverse, and coronal planes for the knee, ankle, hip, pelvis, and foot [179]. Overcoming this limitation could be facilitated by advancements in wearable sensing technologies that can be easily worn and capture gait over extended periods, even outside of gait labs.

- Due to the clinical nature of the studies, the majority of the data has been collected in a laboratory setting under controlled conditions. In Chapters 4 and 5, data from participants walking at self-selected speeds for a predefined distance was utilised, while Chapter 6 utilised data captured under controlled gait speeds on a treadmill. It is important to note that data collected in a laboratory reflects the participants' "walking capacity" or their best performance, whereas data collected in real life reflects their "walking performance," which represents their daily walking habits [180]. Studies have shown that many gait characteristics including gait variability, pace, rhythm, and asymmetry differ when collected in a laboratory compared to outside of it. For instance, when assessed outside the laboratory, stride time and asymmetry are expected to increase while gait speed is expected to decrease. Furthermore, in clinical settings, children with CP often perform better [180]. Therefore, it is necessary for AI models used for gait trajectory prediction to be trained and evaluated on data captured outside of the clinical environment. Some approaches have started collecting field-based data outside of laboratories and structured environments [181]. Advancements in wearable sensors capable of capturing data in diverse environments can also help address this limitation.

- The methods proposed in this study are applications of AI in exoskeleton control for the rehabilitation of children with CP. However, it is important to note that

these models need to be integrated with the mid- and low-level control aspects of an exoskeleton. Several factors, such as the response times of the exoskeletons and their effectiveness in rehabilitation, need to be thoroughly assessed. Although these considerations are limitations of the current study, they present opportunities for future research to address and investigate. Subsequent studies should focus on further exploring the integration of these AI models with the broader control framework of exoskeletons and evaluating their real-world performance in rehabilitation settings.

- In a review on the clinical effectiveness of exoskeleton control strategies for the rehabilitation of individuals with brain injuries, trajectory tracking was found to be the most commonly used for mid-level control (97%) [16]. However, the authors highlighted the difficulty in comparing the effectiveness of different control strategies due to the heterogeneity in experimental protocols across the reviewed studies. Additionally, inconsistencies were observed in the choice of benchmarks used to evaluate the effectiveness of a proposed exoskeleton control mode. These benchmarks could range from conventional gait therapy to unpowered exoskeletons or exoskeletons in zero torque mode. Furthermore, there was wide variability in the outcome measures selected to evaluate the effectiveness of an intervention or treatment. One of the commonly used metrics is walking speed, but other parameters can be considered including biomechanical metrics, step length, set-up timings, adaptation time, and long-term improvements in gait not only during the rehabilitation session but after it as well. Moreover, the effectiveness may be dependent on the specific condition of the individual including factors such as their gross motor function [15, 16]. Given that there is no consensus on the optimal control strategy, the control strategy proposed in this study still requires evaluation in a clinical setting and comparison to other control strategies in a controlled intervention study. Additionally, the proposed approach may need to be combined with an additional control strategy, such as for designing a challenged-based controller.

7.5 Future Directions

Exoskeletons are advancing rapidly and there are several directions for further research and improvement. In the context of the findings of this research, here are some potential directions for future investigations:

- Incorporating Environmental Information: The gait trajectory prediction models developed in this study relied on knowledge of the user's joint positions, specifically using the hip, knee, and joint angles as input features. However, considering information about the external environment, such as terrain type and inclination, can positively impact the user's interaction with the exoskeleton. In a study by Laschowski et al. [129], computer vision techniques were used for classifying the environment based on an onboard wearable camera. Including environmental information for gait trajectory prediction may further enhance human-robot interaction by delivering the most appropriate level of assistance.

- Enhancing Interpretability with Explainable AI (XAI): Deep learning models suffer from the black box problem, making it difficult to understand why certain predictions are generated. Explainable artificial intelligence (XAI) techniques can be applied to enhance the interpretability and transparency of the predictions. Especially in clinical contexts, justifications for predictions are needed to increase confidence in the system and enhance clinical acceptance [21, 30]. For example, when generating adaptive reference gait trajectories for children with Cerebral Palsy, more information needs to be provided on characteristics of the input data that are influencing the result, as well as how much each individual joint pattern contributes to the overall prediction. One of the first examples of implementing XAI in the context of gait analysis was by Horst et al. [30], who used Layer-Wise Relevance Propagation (LRP) to identify which portions of the gait cycle are important for identifying individuals by their gait. LRP highlighted which features of the input to the deep learning model, which include ground reaction forces (GRF) and joint angles of the full body, influence the predictions. Implementing XAI techniques can improve the interpretability, transparency, and

clinical acceptance of gait trajectory prediction models.

- Implementing Reinforcement Learning (RL): Another potential area of research is the use of reinforcement learning (RL) for the generation of adaptive reference trajectories for children with Cerebral Palsy. Reinforcement learning involves an agent and an environment. The agent interacts with the environment, and each interaction is associated with a reward value. The objective of reinforcement learning is for the agent to learn how to interact with the environment through a trial and error process, by rewarding optimal interactions [182]. This is particularly important in the context of exoskeletons which are expected to encounter variable and unpredictable external perturbations and interaction forces between the human and the exoskeletons that can be enhanced by the varying levels of disability of the target users [183]. A study by Luo et al. [184] introduced a reinforcement learning-based walking controller which was virtually tested for its robustness and effectiveness. RL-based approaches eliminate the need for tuning the parameters of the controller for each patient. Investigating the application of RL-based controllers specifically tailored for children with CP would be a promising research direction.

Bibliography

[1] K. Vitrikas, H. Dalton, and D. Breish, "Cerebral palsy: an overview," *American family physician*, vol. 101, no. 4, pp. 213–220, 2020.

[2] "Data and statistics for cerebral palsy," May 2022. [Online]. Available: https://www.cdc.gov/ncbddd/cp/data.html#references

[3] F. Miller, *Cerebral palsy*. Springer Science & Business Media, 2005.

[4] K. W. Krigger, "Cerebral palsy: an overview," *American family physician*, vol. 73, no. 1, pp. 91–100, 2006.

[5] X. Li and K. Arya, "Athetoid cerebral palsy." [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK563160/

[6] M. L. Aisen, D. Kerkovich, J. Mast, S. Mulroy, T. A. Wren, R. M. Kay, and S. A. Rethlefsen, "Cerebral palsy: clinical care and neurological rehabilitation," *The Lancet Neurology*, vol. 10, no. 9, pp. 844–852, 2011.

[7] B. M. Kalkman, L. Bar-On, T. D. O'Brien, and C. N. Maganaris, "Stretching interventions in children with cerebral palsy: why are they ineffective in improving muscle function and how can we better their outcome?" *Frontiers in physiology*, vol. 11, p. 131, 2020.

[8] D. R. Patel, "Therapeutic interventions in cerebral palsy," *The Indian Journal of Pediatrics*, vol. 72, pp. 979–983, 2005.

[9] M. Hunt, L. Everaert, M. Brown, E. Hatzidimitriadou, and K. Desloovere, "Effectiveness of robotic exoskeletons for improving gait in children with cerebral

palsy: A systematic review," *Gait & Posture*, vol. 90, pp. 108–109, 2021, eSMAC 2021 Abstracts.

[10] N. G. Moreau, A. W. Bodkin, K. Bjornson, A. Hobbs, M. Soileau, and K. Lahasky, "Effectiveness of rehabilitation interventions to improve gait speed in children with cerebral palsy: systematic review and meta-analysis," *Physical therapy*, vol. 96, no. 12, pp. 1938–1954, 2016.

[11] M. Sarajchi, M. K. Al-Hares, and K. Sirlantzis, "Wearable lower-limb exoskeleton for children with cerebral palsy: A systematic review of mechanical design, actuation type, control strategy, and clinical evaluation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2021.

[12] V. Kumar, Y. V. Hote, and S. Jain, "Review of exoskeleton: history, design and control," in *2019 3rd International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE)*. IEEE, 2019, pp. 677–682.

[13] L. R. Bunge, A. J. Davidson, B. R. Helmore, A. D. Mavrandonis, T. D. Page, T. R. Schuster-Bayly, and S. Kumar, "Effectiveness of powered exoskeleton use on gait in individuals with cerebral palsy: A systematic review," *PloS one*, vol. 16, no. 5, p. e0252193, 2021.

[14] Y. H. Kwon, J. W. Kwon, and M. H. Lee, "Effectiveness of motor sequential learning according to practice schedules in healthy adults; distributed practice versus massed practice," *Journal of physical therapy science*, vol. 27, no. 3, pp. 769–772, 2015.

[15] M. Hunt, L. Everaert, M. Brown, L. Muraru, E. Hatzidimitriadou, and K. Desloovere, "Effectiveness of robotic exoskeletons for improving gait in children with cerebral palsy: A systematic review," *Gait & Posture*, 2022.

[16] J. de Miguel-Fernández, J. Lobo-Prat, E. Prinsen, J. M. Font-Llagunes, and L. Marchal-Crespo, "Control strategies used in lower limb exoskeletons for gait rehabilitation after brain injury: a systematic review and analysis of clinical effectiveness," *Journal of neuroengineering and rehabilitation*, vol. 20, no. 1, p. 23, 2023.

[17] R. Baud, A. R. Manzoori, A. Ijspeert, and M. Bouri, "Review of control strategies for lower-limb exoskeletons to assist gait," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, no. 1, 2021.

[18] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, 2018.

[19] R. Kolaghassi, M. K. Al-Hares, and K. Sirlantzis, "Systematic review of intelligent algorithms in gait analysis and prediction for lower limb robotic systems," *IEEE Access*, vol. 9, pp. 113 788–113 812, 2021.

[20] A. S. Alharthi, S. U. Yunas, and K. B. Ozanyan, "Deep learning for monitoring of human gait: A review," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 9575–9591, 2019.

[21] E. J. Harris, I.-H. Khoo, and E. Demircan, "A survey of human gait-based artificial intelligence applications," *Frontiers in Robotics and AI*, vol. 8, p. 749274, 2022.

[22] E. Abdulhay, N. Arunkumar, K. Narasimhan, E. Vellaiappan, and V. Venkatraman, "Gait and tremor investigation using machine learning techniques for the diagnosis of parkinson disease," *Future Generation Computer Systems*, vol. 83, pp. 366–373, 2018.

[23] D. Groos, L. Adde, S. Aubert, L. Boswell, R.-A. De Regnier, T. Fjørtoft, D. Gaebler-Spira, A. Haukeland, M. Loennecken, M. Msall *et al.*, "Development and validation of a deep learning method to predict cerebral palsy from spontaneous movements in infants at high risk," *JAMA network open*, vol. 5, no. 7, pp. e2 221 325–e2 221 325, 2022.

[24] C. Yan, B. Zhang, and F. Coenen, "Multi-attributes gait identification by convolutional neural networks," in *2015 8th International Congress on Image and Signal Processing (CISP)*, 2015, pp. 642–647.

[25] Ł. Kidziński, S. Delp, and M. Schwartz, "Automatic real-time gait event detection in children using deep neural networks," *PloS one*, vol. 14, no. 1, p. e0211466, 2019.

[26] Y. Zhang, S. Prasad, A. Kilicarslan, and J. L. Contreras-Vidal, "Multiple kernel based region importance learning for neural classification of gait states from eeg signals," *Frontiers in neuroscience*, vol. 11, p. 170, 2017.

[27] K. Gui, H. Liu, and D. Zhang, "Toward multimodal human-robot interaction to enhance active participation of users in gait rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 2054–2066, 2017.

[28] E. López-Larraz, F. Trincado-Alonso, V. Rajasekaran, S. Pérez-Nombela, A. J. Del-Ama, J. Aranda, J. Minguez, A. Gil-Agudo, and L. Montesano, "Control of an ambulatory exoskeleton with a brain-machine interface for spinal cord injury gait rehabilitation," *Frontiers in neuroscience*, vol. 10, p. 359, 2016.

[29] J.-L. Ren, Y.-H. Chien, E.-Y. Chia, L.-C. Fu, and J.-S. Lai, "Deep learning based motion prediction for exoskeleton robot control in upper limb rehabilitation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5076–5082.

[30] F. Horst, S. Lapuschkin, W. Samek, K.-R. Müller, and W. I. Schöllhorn, "Explaining the unique nature of individual gait patterns with deep learning," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[31] Y. Matsushita, D. T. Tran, H. Yamazoe, and J.-H. Lee, "Recent use of deep learning techniques in clinical applications based on gait: A survey," *Journal of Computational Design and Engineering*, vol. 8, no. 6, pp. 1499–1532, 2021.

[32] 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.

[33] F. Wang, L. Yan, and J. Xiao, "Recognition of the gait phase based on new deep learning algorithm using multisensor information fusion," *Sensors Mater*, vol. 3110, p. 3041, 2019.

[34] B. M. Eskofier, S. I. Lee, J.-F. Daneault, F. N. Golabchi, G. Ferreira-Carvalho, G. Vergara-Diaz, S. Sapienza, G. Costante, J. Klucken, T. Kautz *et al.*, "Recent machine learning advancements in sensor-based mobility analysis: Deep learning for parkinson's disease assessment," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 655–658.

[35] C. Glackin, C. Salge, M. Greaves, D. Polani, S. Slavnić, D. Ristić-Durrant, A. Leu, and Z. Matjačić, "Gait trajectory prediction using gaussian process ensembles," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 628–633.

[36] Z. Zhou, B. Liang, G. Huang, B. Liu, J. Nong, and L. Xie, "Individualized gait generation for rehabilitation robots based on recurrent neural networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 273–281, 2021.

[37] "Research and development prototype for machine augmentation of human strength and endurance hardiman i project." Schenectady, New York: General Electric, 1971.

[38] T. Zhang and H. Huang, "A lower-back robotic exoskeleton: Industrial handling augmentation used to provide spinal support," *IEEE Robotics & Automation Magazine*, vol. 25, no. 2, pp. 95–106, 2018.

[39] A. S. Khan, D. C. Livingstone, C. L. Hurd, J. Duchcherer, J. E. Misiaszek, M. A. Gorassini, P. J. Manns, and J. F. Yang, "Retraining walking over ground in a powered exoskeleton after spinal cord injury: a prospective cohort study to examine functional gains and neuroplasticity," *Journal of neuroengineering and rehabilitation*, vol. 16, no. 1, pp. 1–17, 2019.

[40] Z. F. Lerner, D. L. Damiano, and T. C. Bulea, "A lower-extremity exoskeleton improves knee extension in children with crouch gait from cerebral palsy," *Science translational medicine*, vol. 9, no. 404, 2017.

[41] F. Di Russo, M. Berchicci, R. L. Perri, F. R. Ripani, and M. Ripani, "A passive exoskeleton can push your life up: application on multiple sclerosis patients," *PloS one*, vol. 8, no. 10, p. e77348, 2013.

[42] S. Bai, G. S. Virk, and T. G. Sugar, *Wearable exoskeleton systems: Design, control and applications*. Institution of Engineering and Technology, 2018.

[43] A. B. Zoss, H. Kazerooni, and A. Chu, "Biomechanical design of the berkeley lower extremity exoskeleton (bleex)," *IEEE/ASME Transactions on mechatronics*, vol. 11, no. 2, pp. 128–138, 2006.

[44] C. J. Walsh, K. Endo, and H. Herr, "A quasi-passive leg exoskeleton for load-carrying augmentation," *International Journal of Humanoid Robotics*, vol. 4, no. 03, pp. 487–506, 2007.

[45] K. Suzuki, G. Mito, H. Kawamoto, Y. Hasegawa, and Y. Sankai, "Intention-based walking support for paraplegia patients with robot suit hal," *Advanced Robotics*, vol. 21, no. 12, pp. 1441–1469, 2007.

[46] S. Wang, L. Wang, C. Meijneke, E. Van Asseldonk, T. Hoellinger, G. Cheron, Y. Ivanenko, V. La Scaleia, F. Sylos-Labini, M. Molinari *et al.*, "Design and control of the mindwalker exoskeleton," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 23, no. 2, pp. 277–286, 2014.

[47] H. Herr, "Exoskeletons and orthoses: classification, design challenges and future directions," *Journal of neuroengineering and rehabilitation*, vol. 6, no. 1, pp. 1–9, 2009.

[48] M. C. Faustini, R. R. Neptune, R. H. Crawford, and S. J. Stanhope, "Manufacture of passive dynamic ankle–foot orthoses using selective laser sintering," *IEEE transactions on biomedical engineering*, vol. 55, no. 2, pp. 784–790, 2008.

[49] J. A. Blaya and H. Herr, "Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 12, no. 1, pp. 24–31, 2004.

[50] B. Chen, H. Ma, L.-Y. Qin, F. Gao, K.-M. Chan, S.-W. Law, L. Qin, and W.-H. Liao, "Recent developments and challenges of lower extremity exoskeletons," *Journal of Orthopaedic Translation*, vol. 5, pp. 26–37, 2016.

[51] H. F. Al-Shuka and R. Song, "On low-level control strategies of lower extremity exoskeletons with power augmentation," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*. IEEE, 2018, pp. 63–68.

[52] H. F. Al-Shuka, R. Song, and C. Ding, "On high-level control of power-augmentation lower extremity exoskeletons: Human walking intention," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*. IEEE, 2018, pp. 169–174.

[53] J. Vantilt, K. Tanghe, M. Afschrift, A. K. Bruijnes, K. Junius, J. Geeroms, E. Aertbeliën, F. De Groote, D. Lefeber, I. Jonkers *et al.*, "Model-based control for exoskeletons with series elastic actuators evaluated on sit-to-stand movements," *Journal of neuroengineering and rehabilitation*, vol. 16, no. 1, pp. 1–21, 2019.

[54] H. Zhao, Z. Wang, S. Qiu, Y. Shen, and J. Wang, "Imu-based gait analysis for rehabilitation assessment of patients with gait disorders," in *2017 4th International Conference on Systems and Informatics (ICSAI)*. IEEE, 2017, pp. 622–626.

[55] W. Zeng, C. Yuan, Q. Wang, F. Liu, and Y. Wang, "Classification of gait patterns between patients with parkinson's disease and healthy controls using phase space reconstruction (psr), empirical mode decomposition (emd) and neural networks," *Neural Networks*, vol. 111, pp. 64–76, 2019.

[56] M. Hadizadeh, S. Amri, H. Mohafez, S. A. Roohi, and A. H. Mokhtar, "Gait analysis of national athletes after anterior cruciate ligament reconstruction following three stages of rehabilitation program: Symmetrical perspective," *Gait & posture*, vol. 48, pp. 152–158, 2016.

[57] Y. Wahab and N. A. Bakar, "Gait analysis measurement for sport application based on ultrasonic system," in *2011 IEEE 15th international symposium on consumer electronics (ISCE)*. IEEE, 2011, pp. 20–24.

[58] O. Dehzangi, M. Taherisadr, and R. ChangalVala, "Imu-based gait recognition using convolutional neural networks and multi-sensor fusion," *Sensors*, vol. 17, no. 12, p. 2735, 2017.

[59] S. Tao, X. Zhang, H. Cai, Z. Lv, C. Hu, and H. Xie, "Gait based biometric personal authentication by using mems inertial sensors," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 5, pp. 1705–1712, 2018.

[60] F. Sun, W. Zang, R. Gravina, G. Fortino, and Y. Li, "Gait-based identification for elderly users in wearable healthcare systems," *Information fusion*, vol. 53, pp. 134–144, 2020.

[61] N. Shibuya, B. T. Nukala, A. I. Rodriguez, J. Tsay, T. Q. Nguyen, S. Zupancic, and D. Y. Lie, "A real-time fall detection system using a wearable gait analysis sensor and a support vector machine (svm) classifier," in *2015 Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. IEEE, 2015, pp. 66–67.

[62] I. Birch, T. Birch, and D. Bray, "The identification of emotions from gait," *Science & Justice*, vol. 5, no. 56, pp. 351–356, 2016.

[63] C. Prakash, R. Kumar, and N. Mittal, "Recent developments in human gait research: parameters, approaches, applications, machine learning techniques, datasets and challenges," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 1–40, 2018.

[64] A. Behboodi, N. Zahradka, H. Wright, J. Alesi, and S. C. Lee, "Real-time detection of seven phases of gait in children with cerebral palsy using two gyroscopes," *Sensors*, vol. 19, no. 11, p. 2517, 2019.

[65] W. Hassani, S. Mohammed, H. Rifaï, and Y. Amirat, "Emg based approach for wearer-centered control of a knee joint actuated orthosis," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 990–995.

[66] J. Taborri, E. Palermo, S. Rossi, and P. Cappa, "Gait partitioning methods: A systematic review," *Sensors*, vol. 16, no. 1, p. 66, 2016.

[67] S. Chen, J. Lach, B. Lo, and G.-Z. Yang, "Toward pervasive gait analysis with wearable sensors: A systematic review," *IEEE journal of biomedical and health informatics*, vol. 20, no. 6, pp. 1521–1537, 2016.

[68] R. Caldas, T. Fadel, F. Buarque, and B. Markert, "Adaptive predictive systems applied to gait analysis: A systematic review," *Gait & Posture*, vol. 77, pp. 75–82, 2020.

[69] R. Caldas, M. Mundt, W. Potthast, F. B. de Lima Neto, and B. Markert, "A systematic review of gait analysis methods based on inertial sensors and adaptive algorithms," *Gait & posture*, vol. 57, pp. 204–210, 2017.

[70] G. Bao, L. Pan, H. Fang, X. Wu, H. Yu, S. Cai, B. Yu, and Y. Wan, "Academic review and perspectives on robotic exoskeletons," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 11, pp. 2294–2304, 2019.

[71] E. Alpaydin, *Introduction to Machine Learning.* Cambridge, MA, USA: MIT Press, 2014, ch. Introduction, pp. 1–20.

[72] J.-Y. Jung, W. Heo, H. Yang, and H. Park, "A neural network-based gait phase classification method using sensors equipped on lower limb exoskeleton robots," *Sensors*, vol. 15, no. 11, pp. 27 738–27 759, 2015.

[73] M. D. Sánchez Manchola, M. J. P. Bernal, M. Munera, and C. A. Cifuentes, "Gait phase detection for lower-limb exoskeletons using foot motion data from a single inertial measurement unit in hemiparetic individuals," *Sensors*, vol. 19, no. 13, p. 2988, 2019.

[74] I. Kang, P. Kunapuli, and A. J. Young, "Real-time neural network-based gait phase estimation using a robotic hip exoskeleton," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 1, pp. 28–37, 2020.

[75] A. Hashemi, Y. Lin, W. McNally, B. Laschowski, B. Hosking, A. Wong, and J. McPhee, "Integration of machine learning with dynamics and control: From autonomous cars to biomechatronics," *Transactions- Canadian Society for Mechanical Engineering*, pp. 7–8, 12 2019.

[76] S. K. Goh, H. A. Abbass, K. C. Tan, A. Al-Mamun, N. Thakor, A. Bezerianos, and J. Li, "Spatio–spectral representation learning for electroencephalographic gait-pattern classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 9, pp. 1858–1867, 2018.

[77] M. W. Whittle, *Gait Analysis: An Introduction.* Amsterdam, Netherlands: Elsevier, 1991.

[78] D. Levine, J. Richards, and M. Whittle, *Whittle's Gait Analysis*, 5th ed. Edinburgh, Scotland: Elsevier, 2012.

[79] N. Özkaya, D. Goldsheyder, M. Nordin, and D. Leger, *Fundamentals of Biomechanics: Equilibrium, Motion, and Deformation*, 4th ed. Cham, Switzerland: Springer, 2017.

[80] S. Shafiul Hasan, M. R. Siddiquee, R. Atri, R. Ramon, J. S. Marquez, and O. Bai, "Prediction of gait intention from pre-movement eeg signals: a feasibility study," *Journal of NeuroEngineering and Rehabilitation*, vol. 17, no. 1, pp. 1–16, 2020.

[81] D. Liu, W. Chen, R. Chavarriaga, Z. Pei, and J. d. R. Millán, "Decoding of self-paced lower-limb movement intention: a case study on the influence factors," *Frontiers in Human Neuroscience*, vol. 11, p. 560, 2017.

[82] H. Shibasaki and M. Hallett, "What is the bereitschaftspotential?" *Clinical neurophysiology*, vol. 117, no. 11, pp. 2341–2356, 2006.

[83] B. Gudiño-Mendoza, G. Sanchez-Ante, and J. M. Antelis, "Detecting the intention to move upper limbs from electroencephalographic brain signals," *Computational and mathematical methods in medicine*, vol. 2016, 2016.

[84] Y. He, D. Eguren, J. M. Azorín, R. G. Grossman, T. P. Luu, and J. L. Contreras-Vidal, "Brain–machine interfaces for controlling lower-limb powered robotic systems," *Journal of neural engineering*, vol. 15, no. 2, p. 021004, 2018.

[85] L. Lacourpaille, F. Hug, and A. Nordez, "Influence of passive muscle tension on electromechanical delay in humans," *PloS one*, vol. 8, no. 1, p. e53159, 2013.

[86] Y. Ma, X. Wu, C. Wang, Z. Yi, and G. Liang, "Gait phase classification and assist torque prediction for a lower limb exoskeleton system using kernel recursive least-squares method," *Sensors*, vol. 19, no. 24, p. 5449, 2019.

[87] Y. Hua, J. Fan, G. Liu, X. Zhang, M. Lai, M. Li, T. Zheng, G. Zhang, J. Zhao, and Y. Zhu, "A novel weight-bearing lower limb exoskeleton based on motion intention prediction and locomotion state identification," *IEEE Access*, vol. 7, pp. 37 620–37 638, 2019.

[88] N. Nazmi, M. A. A. Rahman, S.-I. Yamamoto, and S. A. Ahmad, "Walking gait event detection based on electromyography signals using artificial neural network," *Biomedical Signal Processing and Control*, vol. 47, pp. 334–343, 2019.

[89] X. Zhang, S. Sun, C. Li, and Z. Tang, "Impact of load variation on the accuracy of gait recognition from surface emg signals," *Applied Sciences*, vol. 8, no. 9, p. 1462, 2018.

[90] T. Zhen, L. Yan, and P. Yuan, "Walking gait phase detection based on acceleration signals using lstm-dnn algorithm," *Algorithms*, vol. 12, no. 12, p. 253, 2019.

[91] J. D. Farah, N. Baddour, and E. D. Lemaire, "Design, development, and evaluation of a local sensor-based gait phase recognition system using a logistic model decision tree for orthosis-control," *Journal of neuroengineering and rehabilitation*, vol. 16, no. 1, pp. 1–11, 2019.

[92] S. Pasinetti, A. Fornaser, M. Lancini, M. De Cecco, and G. Sansoni, "Assisted gait phase estimation through an embedded depth camera using modified random forest algorithm classification," *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3343–3355, 2019.

[93] P. T. Chinimilli, Z. Qiao, S. M. R. Sorkhabadi, V. Jhawar, I. H. Fong, and W. Zhang, "Automatic virtual impedance adaptation of a knee exoskeleton for personalized walking assistance," *Robotics and Autonomous Systems*, vol. 114, pp. 66–76, 2019.

[94] C.-F. Chen, Z.-J. Du, L. He, Y.-J. Shi, J.-Q. Wang, G.-Q. Xu, Y. Zhang, D.-M. Wu, and W. Dong, "Development and hybrid control of an electrically actuated lower limb exoskeleton for motion assistance," *IEEE Access*, vol. 7, pp. 169 107–169 122, 2019.

[95] W. Huo, S. Mohammed, Y. Amirat, and K. Kong, "Fast gait mode detection and assistive torque control of an exoskeletal robotic orthosis for walking assistance," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1035–1052, 2018.

[96] C. Chen, X. Wu, D.-x. Liu, W. Feng, and C. Wang, "Design and voluntary motion intention estimation of a novel wearable full-body flexible exoskeleton robot," *Mobile Information Systems*, vol. 2017, 2017.

[97] K. Tanghe, F. De Groote, D. Lefeber, J. De Schutter, and E. Aertbeliën, "Gait trajectory and event prediction from state estimation for exoskeletons during gait," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 211–220, 2019.

[98] J. Song, A. Zhu, Y. Tu, Y. Wang, M. A. Arif, H. Shen, Z. Shen, X. Zhang, and G. Cao, "Human body mixed motion pattern recognition method based on multi-source feature parameter fusion," *Sensors*, vol. 20, no. 2, p. 537, 2020.

[99] M. Islam and E. T. Hsiao-Wecksler, "Detection of gait modes using an artificial neural network during walking with a powered ankle-foot orthosis," *Journal of Biophysics*, vol. 2016, 2016.

[100] F. Wang, L. Yan, and J. Xiao, "Human gait recognition system based on support vector machine algorithm and using wearable sensors," *Sensors and Materials*, vol. 31, no. 4, pp. 1335–1349, 2019.

[101] A. C. Villa-Parra, D. Delisle-Rodriguez, T. Botelho, J. J. V. Mayor, A. L. Delis, R. Carelli, A. Frizera Neto, and T. F. Bastos, "Control of a robotic knee exoskeleton for assistance and rehabilitation based on motion intention from semg," *Research on Biomedical Engineering*, vol. 34, pp. 198–210, 2018.

[102] A. Parri, K. Yuan, D. Marconi, T. Yan, S. Crea, M. Munih, R. M. Lova, N. Vitiello, and Q. Wang, "Real-time hybrid locomotion mode recognition for lower

limb wearable robots," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2480–2491, 2017.

[103] D. Novak, P. Reberšek, S. M. M. De Rossi, M. Donati, J. Podobnik, T. Beravs, T. Lenzi, N. Vitiello, M. C. Carrozza, and M. Munih, "Automated detection of gait initiation and termination using wearable sensors," *Medical engineering & physics*, vol. 35, no. 12, pp. 1713–1720, 2013.

[104] Y. Zheng, Q. Song, J. Liu, Q. Song, and Q. Yue, "Research on motion pattern recognition of exoskeleton robot based on multimodal machine learning model," *Neural Computing and Applications*, vol. 32, no. 7, pp. 1869–1877, 2020.

[105] K. Gui, H. Liu, and D. Zhang, "A practical and adaptive method to achieve emg-based torque estimation for a robotic exoskeleton," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 2, pp. 483–494, 2019.

[106] B. Xiong, N. Zeng, H. Li, Y. Yang, Y. Li, M. Huang, W. Shi, M. Du, and Y. Zhang, "Intelligent prediction of human lower extremity joint moment: An artificial neural network approach," *IEEE Access*, vol. 7, pp. 29 973–29 980, 2019.

[107] B. Xiong, N. Zeng, Y. Li, M. Du, M. Huang, W. Shi, G. Mao, and Y. Yang, "Determining the online measurable input variables in human joint moment intelligent prediction based on the hill muscle model," *Sensors*, vol. 20, no. 4, p. 1185, 2020.

[108] P. Kutilek and B. Farkasova, "Prediction of lower extremities' movement by angle-angle diagrams and neural networks." *Acta of Bioengineering & Biomechanics*, vol. 13, no. 2, 2011.

[109] P. Kutilek and S. Viteckova, "Prediction of lower extremity movement by cyclograms," *Acta Polytechnica*, vol. 52, no. 1, 2012.

[110] O. Mazumder, A. S. Kundu, P. K. Lenka, and S. Bhaumik, "Multi-channel fusion based adaptive gait trajectory generation using wearable sensors," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 3, pp. 335–351, 2017.

[111] J.-W. Lee and G.-K. Lee, "Gait angle prediction for lower limb orthotics and prostheses using an emg signal and neural networks," *International Journal of Control, Automation, and Systems*, vol. 3, no. 2, pp. 152–158, 2005.

[112] H. Xie, G. Li, X. Zhao, and F. Li, "Prediction of limb joint angles based on multi-source signals by gs-grnn for exoskeleton wearer," *Sensors*, vol. 20, no. 4, p. 1104, 2020.

[113] J. Wang, L. Wang, S. M. Miran, X. Xi, and A. Xue, "Surface electromyography based estimation of knee joint angle by using correlation dimension of wavelet coefficient," *IEEE Access*, vol. 7, pp. 60 522–60 531, 2019.

[114] M. A. Gomes, G. L. M. Silveira, and A. A. Siqueira, "Gait pattern adaptation for an active lower-limb orthosis based on neural networks," *Advanced Robotics*, vol. 25, no. 15, pp. 1903–1925, 2011.

[115] X. Wu, D.-X. Liu, M. Liu, C. Chen, and H. Guo, "Individualized gait pattern generation for sharing lower limb exoskeleton robot," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1459–1470, 2018.

[116] A. M. Boudali, P. J. Sinclair, and I. R. Manchester, "Predicting transitioning walking gaits: Hip and knee joint trajectories from the motion of walking canes," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 9, pp. 1791–1800, 2019.

[117] H. Vallery, E. H. Van Asseldonk, M. Buss, and H. Van Der Kooij, "Reference trajectory generation for rehabilitation robots: complementary limb motion estimation," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 17, no. 1, pp. 23–30, 2008.

[118] M. Hassan, H. Kadone, T. Ueno, Y. Hada, Y. Sankai, and K. Suzuki, "Feasibility of synergy-based exoskeleton robot control in hemiplegia," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 6, pp. 1233–1242, 2018.

[119] R. Xu, N. Jiang, N. Mrachacz-Kersting, C. Lin, G. A. Prieto, J. C. Moreno, J. L. Pons, K. Dremstrup, and D. Farina, "A closed-loop brain–computer interface

triggering an active ankle–foot orthosis for inducing cortical neural plasticity," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 7, pp. 2092–2101, 2014.

[120] P. Silsupadol, K. Teja, and V. Lugade, "Reliability and validity of a smartphone-based assessment of gait parameters across walking speed and smartphone locations: Body, bag, belt, hand, and pocket," *Gait & posture*, vol. 58, pp. 516–522, 2017.

[121] S. Qiu, Z. Wang, H. Zhao, and H. Hu, "Using distributed wearable sensors to measure and evaluate human lower limb motions," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 939–950, 2016.

[122] A. R. Anwary, H. Yu, and M. Vassallo, "Optimal foot location for placing wearable imu sensors and automatic feature extraction for gait analysis," *IEEE Sensors Journal*, vol. 18, no. 6, pp. 2555–2567, 2018.

[123] P. Slade, R. Troutman, M. J. Kochenderfer, S. H. Collins, and S. L. Delp, "Rapid energy expenditure estimation for ankle assisted and inclined loaded walking," *Journal of neuroengineering and rehabilitation*, vol. 16, no. 1, pp. 1–10, 2019.

[124] Y. Gao and Y. Cui, "Deep transfer learning for reducing health care disparities arising from biomedical data inequality," *Nature communications*, vol. 11, no. 1, pp. 1–8, 2020.

[125] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[126] M. Seiffert, F. Holstein, R. Schlosser, and J. Schiller, "Next generation cooperative wearables: Generalized activity assessment computed fully distributed within a wireless body area network," *IEEE Access*, vol. 5, pp. 16 793–16 807, 2017.

[127] "Automatic real-time gait event detection in children using deep neural networks," https://simtk.org/frs/?group_id=1946.

[128] J. Camargo, A. Ramanathan, W. Flanagan, and A. Young, "A comprehensive, open-source dataset of lower limb biomechanics in multiple conditions of stairs,

ramps, and level-ground ambulation and transitions," *Journal of Biomechanics*, vol. 119, p. 110320, 2021.

[129] B. Laschowski, W. McNally, A. Wong, and J. McPhee, "Environment classification for robotic leg prostheses and exoskeletons using deep convolutional neural networks," *Frontiers in Neurorobotics*, vol. 15, 2022.

[130] B. Chen, H. Ma, L.-Y. Qin, F. Gao, K.-M. Chan, S.-W. Law, L. Qin, and W.-H. Liao, "Recent developments and challenges of lower extremity exoskeletons," *Journal of Orthopaedic Translation*, vol. 5, pp. 26–37, 2016, special Issue: Orthopaedic Biomaterials and Devices.

[131] D.-X. Liu, X. Wu, C. Wang, and C. Chen, "Gait trajectory prediction for lower-limb exoskeleton based on deep spatial-temporal model (dstm)," in *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2017, pp. 564–569.

[132] A. Zaroug, D. T. H. Lai, K. Mudie, and R. Begg, "Lower limb kinematics trajectory prediction using long short-term memory neural networks," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 362, 2020.

[133] B. Su and E. M. Gutierrez-Farewik, "Gait trajectory and gait phase prediction based on an lstm network," *Sensors*, vol. 20, no. 24, 2020.

[134] V. Hernandez, D. Dadkhah, V. Babakeshizadeh, and D. Kulić, "Lower body kinematics estimation from wearable sensors for walking and running: A deep learning approach," *Gait and Posture*, vol. 83, pp. 185–193, 2021.

[135] L. Jia, Q. Ai, W. Meng, Q. Liu, and S. Q. Xie, "Individualized gait trajectory prediction based on fusion lstm networks for robotic rehabilitation training," in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2021, pp. 988–993.

[136] A. Zaroug, A. Garofolini, D. T. H. Lai, K. Mudie, and R. Begg, "Prediction of gait trajectories based on the long short term memory neural networks," *PLOS ONE*, vol. 16, no. 8, pp. 1–19, 08 2021.

[137] C. Zhu, Q. Liu, W. Meng, Q. Ai, and S. Q. Xie, "An attention-based cnn-lstm model with limb synergy for joint angles prediction," in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2021, pp. 747–752.

[138] C. Kirtley, *Clinical gait analysis: Theory and practice.* Churchill Livingstone Elsevier, 2006.

[139] G. Steinwender, V. Saraph, S. Scheiber, E. B. Zwick, C. Uitz, and K. Hackl, "Intrasubject repeatability of gait analysis data in normal and spastic children," *Clinical Biomechanics*, vol. 15, no. 2, pp. 134–139, 2000.

[140] Y. Moon, J. Sung, R. An, M. E. Hernandez, and J. J. Sosnoff, "Gait variability in people with neurological disorders: A systematic review and meta-analysis," *Human Movement Science*, vol. 47, pp. 197–208, 2016.

[141] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[142] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[143] E. Gieysztor, M. Kowal, and M. Paprocka-Borowicz, "Gait parameters in healthy preschool and school children assessed using wireless inertial sensor," *Sensors*, vol. 21, no. 19, 2021.

[144] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, p. 107398, 2021.

[145] L. Moreira, J. Figueiredo, J. P. Vilas-Boas, and C. P. Santos, "Kinematics, speed, and anthropometry-based ankle joint torque estimation: A deep learning regression approach," *Machines*, vol. 9, no. 8, 2021.

[146] D. D. Molinaro, I. Kang, J. Camargo, M. C. Gombolay, and A. J. Young, "Subject-independent, biological hip moment estimation during multimodal over-

ground ambulation using deep learning," *IEEE Transactions on Medical Robotics and Bionics*, vol. 4, no. 1, pp. 219–229, 2022.

[147] H. Kawamoto, T. Hayashi, T. Sakurai, K. Eguchi, and Y. Sankai, "Development of single leg version of hal for hemiplegia," in *2009 Annual international conference of the IEEE engineering in medicine and biology society.* IEEE, 2009, pp. 5038–5043.

[148] D. Eguren, M. Cestari, T. P. Luu, A. Kilicarslan, A. Steele, and J. L. Contreras-Vidal, "Design of a customizable, modular pediatric exoskeleton for rehabilitation and mobility," in *2019 IEEE international conference on systems, man and cybernetics (SMC).* IEEE, 2019, pp. 2411–2416.

[149] M. Maggu, R. Udasi, and D. Nikitina, "Designing exoskeletons for children: Overcoming challenge associated with weight-bearing and risk of injury," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 39–39.

[150] C. Bayón, T. Martín-Lorenzo, B. Moral-Saiz, Ó. Ramírez, Á. Pérez-Somarriba, S. Lerma-Lara, I. Martínez, and E. Rocon, "A robot-based gait training therapy for pediatric population with cerebral palsy: goal setting, proposal and preliminary clinical implementation," *Journal of neuroengineering and rehabilitation*, vol. 15, no. 1, pp. 1–15, 2018.

[151] M. A. H. Mohd Adib, S. Y. Han, P. R. Ramani, L. J. You, L. M. Yan, I. Mat Sahat, and N. H. Mohd Hasni, "Restoration of kids leg function using exoskeleton robotic leg (exroleg) device," in *Proceedings of the 10th National Technical Seminar on Underwater System Technology 2018.* Springer, 2019, pp. 335–342.

[152] F. Patané, S. Rossi, F. Del Sette, J. Taborri, and P. Cappa, "Wake-up exoskeleton to assist children with cerebral palsy: design and preliminary evaluation in level walking," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 7, pp. 906–916, 2017.

[153] R. Kolaghassi, M. K. Al-Hares, G. Marcelli, and K. Sirlantzis, "Performance of deep learning models in forecasting gait trajectories of children with neurological disorders," *Sensors*, vol. 22, no. 8, p. 2969, 2022.

[154] M. F. B. Miskon and M. B. A. J. Yusof, "Review of trajectory generation of exoskeleton robots," in *2014 IEEE international symposium on robotics and manufacturing automation (ROMA)*. IEEE, 2014, pp. 12–17.

[155] R. L. McGrath, M. Pires-Fernandes, B. Knarr, J. S. Higginson, and F. Sergi, "Toward goal-oriented robotic gait training: The effect of gait speed and stride length on lower extremity joint torques," in *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2017, pp. 270–275.

[156] B. Ren, Z. Zhang, C. Zhang, and S. Chen, "Motion trajectories prediction of lower limb exoskeleton based on long short-term memory (lstm) networks," in *Actuators*, vol. 11, no. 3. MDPI, 2022, p. 73.

[157] Z.-Q. Ling, G.-Z. Cao, Y.-P. Zhang, H.-R. Cheng, B.-B. He, and S.-B. Cao, "Real-time knee joint angle estimation based on surface electromyograph and back propagation neural network," in *2021 18th International Conference on Ubiquitous Robots (UR)*. IEEE, 2021, pp. 256–263.

[158] S. A. A. Moosavian, A. Kiani, V. Akbari, M. Nabipour, and S. Ghanaat, "Robowalk trajectory planning based on the human gait prediction using lstm," in *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*. IEEE, 2021, pp. 433–438.

[159] Z. Guo, J. Ye, S. Zhang, L. Xu, G. Chen, X. Guan, Y. Li, and Z. Zhang, "Effects of individualized gait rehabilitation robotics for gait training on hemiplegic patients: Before-after study in the same person," *Frontiers in Neurorobotics*, vol. 15, 2022.

[160] M. Endo, K. L. Poston, E. V. Sullivan, L. Fei-Fei, K. M. Pohl, and E. Adeli, "Gaitforemer: Self-supervised pre-training of transformers via human motion forecasting for few-shot gait impairment severity estimation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2022, pp. 130–139.

[161] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[162] D. Hutchins, I. Schlag, Y. Wu, E. Dyer, and B. Neyshabur, "Block-recurrent transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 248–33 261, 2022.

[163] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[164] T. Giorgino, "Computing and visualizing dynamic time warping alignments in r: the dtw package," *Journal of statistical Software*, vol. 31, pp. 1–24, 2009.

[165] C. A. Fukuchi and M. Duarte, "A prediction method of speed-dependent walking patterns for healthy individuals," *Gait & Posture*, vol. 68, pp. 280–284, 2019.

[166] G. M. Bryan, P. W. Franks, S. Song, A. S. Voloshina, R. Reyes, M. P. O'Donovan, K. N. Gregorczyk, and S. H. Collins, "Optimized hip-knee-ankle exoskeleton assistance at a range of walking speeds," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, pp. 1–12, 2021.

[167] C. A. Fukuchi, R. K. Fukuchi, and M. Duarte, "Effects of walking speed on gait biomechanics in healthy participants: a systematic review and meta-analysis," *Systematic reviews*, vol. 8, pp. 1–11, 2019.

[168] A. Van Hamme, A. El Habachi, W. Samson, R. Dumas, L. Cheze, and B. Dohin, "Gait parameters database for young children: The influences of age and walking speed," *Clinical biomechanics*, vol. 30, no. 6, pp. 572–577, 2015.

[169] A. Delval, J. Salleron, J.-L. Bourriez, S. Bleuse, C. Moreau, P. Krystkowiak, L. Defebvre, P. Devos, and A. Duhamel, "Kinematic angular parameters in pd: reliability of joint angle curves and comparison with healthy subjects," *Gait & posture*, vol. 28, no. 3, pp. 495–501, 2008.

[170] S. Marrocco, L. D. Crosby, I. C. Jones, R. F. Moyer, T. B. Birmingham, and K. K. Patterson, "Knee loading patterns of the non-paretic and paretic legs during post-stroke gait," *Gait & posture*, vol. 49, pp. 297–302, 2016.

[171] A. Patoz, T. Lussiana, C. Gindre, and L. Mourot, "Predicting temporal gait kinematics: Anthropometric characteristics and global running pattern matter," *Frontiers in Physiology*, vol. 11, p. 625557, 2021.

[172] C. Zou, R. Huang, H. Cheng, and J. Qiu, "Learning gait models with varying walking speeds," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 183–190, 2020.

[173] H. Han, H. An, H. Ma, and Q. Wei, "Variable speed trajectory prediction method of dynamic prosthesis based on improved seqgan," in *2021 China Automation Congress (CAC)*. IEEE, 2021, pp. 2628–2632.

[174] K. R. Embry, D. J. Villarreal, R. L. Macaluso, and R. D. Gregg, "Modeling the kinematics of human locomotion over continuously varying speeds and inclines," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 26, no. 12, pp. 2342–2350, 2018.

[175] L. Lu, S. Liu, Z. Zhou, J. Sun, A. Melendez-Calderon, and Z. Guo, "Continuous gait phase recognition and prediction using lstm network across walking speeds and inclinations," in *2022 International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2022, pp. 101–106.

[176] T. Öberg, A. Karsznia, and K. Öberg, "Basic gait parameters: reference data for normal subjects, 10-79 years of age," *Journal of rehabilitation research and development*, vol. 30, pp. 210–210, 1993.

[177] S. Al-Obaidi, J. C. Wall, A. Al-Yaqoub, and M. Al-Ghanim, "Basic gait parameters: A comparison of reference data for normal subjects 20 to 29 years of age from kuwait and scandinavia," *Journal of rehabilitation research and development*, vol. 40, no. 4, pp. 361–366, 2003.

[178] R. Kolaghassi, G. Marcelli, and K. Sirlantzis, "Deep learning models for stable gait prediction applied to exoskeleton reference trajectories for children with cerebral palsy," *IEEE Access*, vol. 11, pp. 31 962–31 976, 2023.

[179] A. Nieuwenhuys, E. Papageorgiou, S.-H. Schless, T. De Laet, G. Molenaers, and K. Desloovere, "Prevalence of joint gait patterns defined by a delphi consensus

study is related to gross motor function, topographical classification, weakness, and spasticity, in children with cerebral palsy," *Frontiers in Human Neuroscience*, vol. 11, 2017.

[180] L. Carcreff, C. N. Gerber, A. Paraschiv-Ionescu, G. De Coulon, C. J. Newman, K. Aminian, and S. Armand, "Comparison of gait characteristics between clinical and daily life settings in children with cerebral palsy," *Scientific reports*, vol. 10, no. 1, pp. 1–11, 2020.

[181] B. Russell, A. McDaid, W. Toscano, and P. Hume, "Moving the lab into the mountains: A pilot study of human activity recognition in unstructured environments," *Sensors*, vol. 21, no. 2, 2021.

[182] Z. Ding, Y. Huang, H. Yuan, and H. Dong, *Introduction to Reinforcement Learning.* Singapore: Springer Singapore, 2020, pp. 47–123.

[183] S. Luo, G. Androwis, S. Adamovich, H. Su, E. Nunez, and X. Zhou, "Reinforcement learning and control of a lower extremity exoskeleton for squat assistance," *Frontiers in Robotics and AI*, vol. 8, 2021.

[184] S. Luo, G. Androwis, S. Adamovich, E. Nunez, H. Su, and X. Zhou, "Robust walking control of a lower limb rehabilitation exoskeleton coupled with a musculoskeletal model via deep reinforcement learning," *Journal of neuroengineering and rehabilitation*, vol. 20, no. 1, pp. 1–19, 2023.

[185] *Encyclopedia of Machine Learning.*, ch. Supervised Learning.

[186] C. Sammut and G. I. . Webb, *Encyclopedia of Machine Learning.* Boston, MA, USA: Springer, 2010, ch. Unsupervised Learning.

[187] P. Stone, *Encyclopedia of Machine Learning.* Boston, MA, USA: Springer, 2010, ch. Reinforcment learning.

[188] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: Association for Computing Machinery, 1992, pp. 144—152.

[189] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[190] N. Cristianini and E. Ricci, "Support vector machines. encyclopedia of algorithms," *Springer*, vol. 10, p. 928–932, 2008.

[191] W. Ertel, *Introduction to Artificial Intelligence*. London, U.K.: Springer, 2011.

[192] J. Fürnkranz, P. Chan, S. Craw, C. Sammut, W. Uther, A. Ratnaparkhi, X. Jin, J. Han, Y. Yang, K. Morik *et al.*, *Encyclopedia of machine learning*. Boston, MA, USA: Springer, 2010, ch. Decision Tree, pp. 263–267.

[193] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81–106, 1986.

[194] S. L. Salzberg, "C4. 5: Programs for machine learning," *Machine Learning*, vol. 16, no. 3, pp. 235–240, 1994.

[195] C. Buckner and J. Garson, "Connectionism," in *The Stanford Encyclopedia of Philosophy*, Fall 2019 ed. Metaphysics Research Lab, Stanford University, 2019.

[196] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[197] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[198] R. Miikkulainen, *Encyclopedia of machine learning*. Boston, MA, USA: Springer, 2010, ch. Topology of a neural network, pp. 988—989.

[199] S. I. Gallant, *Neural network learning and expert systems*. Cambridge, MA, USA: MIT press, 1993.

[200] S. Mirjalili, "Evolutionary radial basis function networks," *Evolutionary Algorithms and Neural Networks: Theory and Applications*, pp. 105–139, 2019.

[201] P. Munro, *Encyclopedia of machine learning*. Boston, MA, USA: Springer, 2010, ch. Backpropagation, p. 73.

[202] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, contour and grouping in computer vision*. Springer, 1999, pp. 319–345.

Appendices

A Machine Learning Algorithms

An algorithm is a set of procedures to transform an input into an output. When the procedures are unknown, and cannot be explicitly programmed, it is possible to approximate the transformation using machine learning algorithms. Machine learning algorithms learn patterns between inputs and outputs, based on large amounts of data in a specific subject domain. Learning is achieved by optimising a pre-selected performance metric, such as the accuracy of output prediction. Machine learning algorithms can be broadly categorised based on their style of learning [71]. The three main categories are supervised, unsupervised and reinforcement learning. Supervised learning algorithms require input and target output data. They continuously compare the algorithm's generated output with the target output until the error between them is minimized [185]. Meanwhile, unsupervised learning algorithms do not require target output data. They aim to find inherent patterns within the structure of the data [186]. Reinforcement learning also does not require target output data and learns through a reward system [187]. Some of the machine learning algorithms include:

A.1 Support Vector Machine

Support Vector Machine (SVM) is an algorithm used for both classification and regression problems. This supervised machine learning algorithm can separate linearly separable classes or features using a hyperplane. In the case of two classes that can be separated by a linear line (hyperplane), as in Figure 1, the optimal hyperplane is chosen to be the one that maximizes the distance between itself and the classes. The objective is to maximize the margin. If the classes are not linearly separable, they can be transformed into higher dimensions where they are linearly separable, or kernels can be used [188–191].

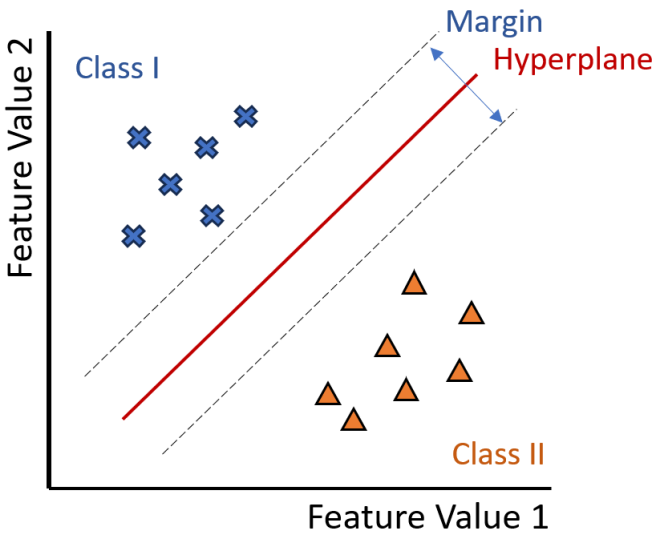

Figure 1: Illustration of the Support Vector Machine that uses a hyperplane to separate linearly separable classes or features.

A.2 Decision Trees

Decision trees, illustrated in Figure 2, are another type of supervised machine-learning algorithm used for classification. They have a hierarchical tree-like structure that starts with a root node and each node is divided into multiple branches based on an attribute value. These nodes are initially considered as 'impure' and continue dividing based on the attribute's value until they reach the leaf node, an indivisible 'pure' node that represents a single class [191,192]. ID3 [193] and C4.5 [194] are popular examples of decision trees.

A.3 Neural Networks

Neural networks are connectionist networks, heavily inspired by the neurons of the brain [195]. The history of neural networks began with McCulloch and Pitts mathematical approximation of a neuron in 1943 [196], followed by Rosenblatt perceptron in the 1960s [197]. Neural networks consist of input, hidden and output layers. Each layer contains nodes, also called neurons which perform a non-linear mathematical operation on input data. The mathematical operation is known as an activation function and the sigmoid function or tanh function are some examples. Each node has an activation value that is dependent on the input to the node, and the weight assigned to it. The values of the weights are adaptive and change as the neural network is learning. If the input influences the generation of the output, a high weight is chosen, leading to a high activation value. Otherwise, a low weight is chosen. The multilayer perceptron (MLP)

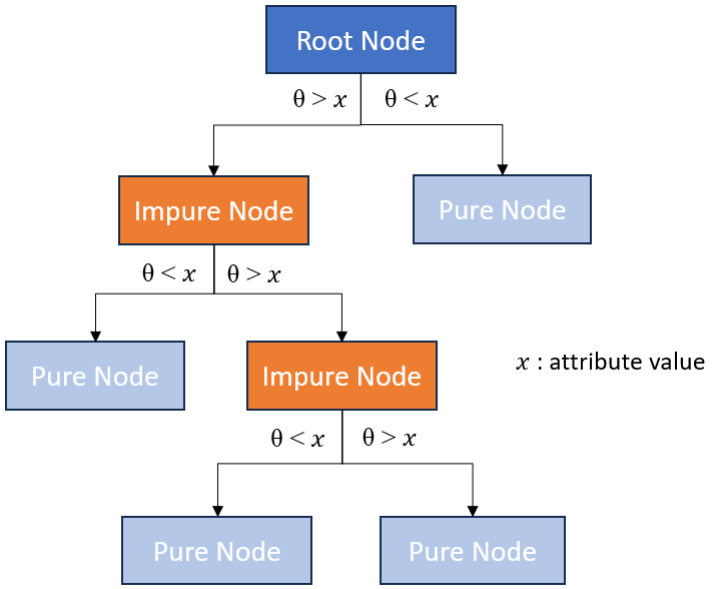

Figure 2: A Simplified Decision Tree that separates the classes based on an attribute value.

is a fully connected feed-forward neural network, where a node in a particular layer is connected to all the nodes in the previous layer and all the nodes in the preceding layer. Nodes within the same layer are not connected. This topology allows for parallel computations [191, 198, 199].

The radial basis function neural network (RBFNN) is a type of neural network that differs from the multilayer perceptron (MLP) in several ways. Unlike the MLP which can have multiple hidden layers, the RBFNN can only have one hidden layer. The computation performed by the nodes in RBFNN is also distinct. The nodes calculate the Euclidean distance between the input and pre-defined prototypes. These prototypes are typically pre-set using an unsupervised algorithm such as the k-Means algorithm. To calculate the output of a node, a nonlinear Gaussian function is used as the activation function. The closer the data point is to the prototype, the greater its influence on the output, since the output of the Gaussian function would be close to one. Conversely, the further away the data point is from the prototype, the lower its influence on the output since the output of the Gaussian function would be close to zero [200].

Both of the aforementioned algorithms are supervised learning algorithms that rely on weights and parameters. The values of these weights are selected in a way to maximize the performance of an algorithm. Backpropagation is commonly used to update the values of the weights. It begins by calculating the error or cost between the algorithm's prediction and the target output. Errors that are often calculated include cross-entropy errors or squared errors. Afterwards, the derivative of the error with respect to the chosen weights is computed. The objective is to

minimize this error, and this is achieved by iteratively adjusting the weights based on the calculated derivative until a satisfactory performance is achieved. This process is a form of optimisation, known as gradient descent [201].

A.4 Deep Neural Networks

Deep neural networks are essentially neural networks with many hidden layers. The number of hidden layers determines the model's depth [141].

Convolutional neural networks (CNNs) are a type of deep neural networks commonly utilised for images. The main operation in a CNN is the convolution operation. A kernel, smaller than the size of the original image, is applied to the image and slides across it. The kernel performs the convolution operation with each portion of the image, producing feature maps. Feature maps are analogous to hidden layers in ANNs. Unlike hidden layers in ANNs which have nodes that connect to every single input, in CNNs, a group of inputs the size of the kernel would be mapped to a single point on the feature map. This is one of the special features of a CNN, known as sparsity of connections, which reduces the number of parameters in the model leading to more efficient memory storage space and computational power requirements. Another notable feature of CNNs is parameter sharing. To produce a single feature map, the same set of weights is used across the entire image. The weights are only adjusted when producing feature maps that extract different features. In addition to convolution layers, there are also other pooling layers, which downsample feature maps. After several alternating convolution and pooling operations, the final feature map is flattened into a fully connected hidden layer to generate the output [141, 202].

While CNNs are effective for 2D input data, another category of deep learning algorithms that are effective with 1D sequential data, such as time series, are Recurrent neural networks (RNNs). RNNs have recurrent connections allowing previous outputs to be used for calculating current outputs. Long short-term memory (LSTM) algorithm is a type of gated RNN. Its architecture consists of cells with an input, an output, and a forget gate. LSTMs have the capability of learning long-term dependencies, meaning that they can generate output depending on input data that happened much earlier in time [141].

B Systematic Literature Review Tables

B.1 Gait Phase

Table 1: Existing literature on gait phases

Authors	Model	Phase Granularity	Performance Metric	Value of Performance Metric	Optimization Algorithm	Sensors	Sample	Purpose
Jung et al [72]	MLP (a) NARX (b)	2	CSR	97.75% (a) 97.63% (b) (offline) 90.75% (a) 91.93% (b) (online)	Back-propagation	IMUs absolute encoders incremental encoders force plates (for segmentation)	10 healthy (7 offline training + validation, 3 online validation)	Stroke patient rehabilitation
Kang et al [74]	NN	% of the gait cycle	RMSE (for semi-dependent model)	5.07 ± 0.49% (steady-state) 5.22 ± 0.81% (dynamic)	SGD optimizer with Nesterov momentum	absolute encoders (on actuators) IMUs (thigh and trunk) FSRs (for segmentation)	8 healthy (initial data collection) 10 healthy (for validation)	Bilateral hip exoskeleton
Hua et al [87]	ANFIS	2	-	-	-	Plantar pressure	-	heavy load lifting
Nazmi et al [88]	MLPNN	2	accuracy (group 2 features and LM algorithm) mean absolute time difference compared to FSRs (for unlearned data)	87.4% 35±25 ms (heel strike) 49±15 ms (toe-off)	Levenberg Marquardt (a) scaled conjugate gradient (b)	EMG (tibialis anterior, gastrocnemius medialis) foot pressure sensor (evaluation only)	8 healthy	-
Zhang et al [89]	BPNN (a) kNN (b) SVM (c)	5	accuracy	91.81 ± 3.69% (intra-load) (a) 69.42 ± 7.86% (inter-load) (a)	-	EMG (tensor fasciae latae, vastus medialis, semitendinosus adductor longus) Camera (for segmenting gait cycle)	10 healthy	-

Zhen et al [90]	LSTM-DNN (a) LSTM (b) KNN (c) SVM (d)	2	Accuracy (1 m/s velocity) F-score (1 m/s velocity)	91.8% (a) 86.8% (b) 69.0% (c) 71.0% (d) 92.0% (a) 86.4% (b) 70.0% (c) 72.3% (d)	AdaGrad	IMUs (thigh, instep, calf)	4 healthy	-
Wang et al [33]	DM-CNN (a) KNN (b) N-HMM (c) HMM (d)	4	Accuracy Precision Rate Recall Rate	97.1% (a) 88.5% (b) 96.2% (c) 92.3% (d) 95.9% (a) 81.5% (b) 95.5% (c) 92.0% (d) 94.5% (a) 82.5% (b) 93.5% (c) 90.5% (d)	Back-propagation	pressure sensors IMUs (calf and thigh)	10 healthy	-
Farah et al [91]	DT+TSVC	4	Accuracy (validation set) F-score (validation set)	98.61% 0.97	-	Computer-Assisted Rehabilitation Environment (CAREN-extended) containing: Force-plates Motion Capture System	30 healthy (training) + 12 healthy (validation)	Stance-control knee-ankle-foot orthoses for knee-collapse prevention
Pasinetti et al [92]	RF (a) Sigma-z RF (b)	2	Accuracy	81% (a) 87.3% (b)	-	time of flight cameras	4	walking in in-door/outdoor environments (with crutches)
Tanghe et al [97]	PPCA	4	Median Error Without exoskeleton With exoskeleton	<9ms (all events) 15ms (initial contact) 33ms (toe-off)	-	Vicon cameras split belt treadmill for GRF (for segmentation)	28 healthy (without exo) + 5 healthy (with exo)	Exoskeleton that uses feedforward allowing control modes to transition more smoothly
Chinimilli et al [93]	Fuzzy logic	4	-	-	-	smart shoes IMUs (thigh and shank + not used for gait phases) EMG (for testing only) Motion capture	3 healthy	Knee assistive device with automatic virtual impedance modulation

Chen et al [94]	Fuzzy logic	4	-	-	-	foot pressure sensors	-	Locomotion assistance using hybrid control
Huo et al [95]	Fuzzy logic	4	-	-	-	Foot pressure sensor (GRF) IMUs Absolute encoders (hip, knee, and ankle)	4 healthy	Walking assistance using hybrid power assistive control
Chen et al [96]	kNN	8	Correct rate of phase classification	95.32%	-	Joint angle sensor (hip, knee and ankle) plantar pressure sensor	10 healthy	Locomotion assistance full body exoskeleton
Manchola et al [73]	HMM	4	accuracy (SST) accuracy (SPT)	81.44% (H) 78.06% (P) 76.91% (H) 76.36% (P)	-	IMU (instep) FSR per foot (reference and testing only)	18 (9 healthy, 9 hemi-paretic)	exoskeleton for hemi-paretic patients
Ma et al [86]	KRLS (a) SVM (b) MLPNN (c)	4	accuracy (10-fold cross validation)	86.26% (a) 83.29% (b) 83.23% (c)	-	goniometer (hip and knee joint) foot pressure sensor (for segmentation)	10 healthy	Walking assistance for paraplegics + rehabilitation of stroke patients

B.2 Locomotion Mode

Table 2: Existing literature on locomotion modes

Authors	**Model**	**Number of Modes**	**Modes**	**Performance Metric**	**Value of Performance Metric**	**Optimization Algorithm**	**Sensors**	**Sample**	**Purpose**
Song et al [98]	BPNN	15	sitting standing running level walking (multiple paces and with weight) etc. (refer to paper for exhaustive list)	accuracy (single-mode) accuracy (multi-mode)	98.28% 92.7% (group 1) 97.4% (group 2)	-	IMUs per leg (foot, calf and thigh) foot pressure sensors Vicon motion capture (for normalization and validation of sensors)	healthy	-

Islam et al [99]	NN	3	level walking ascending stairs/ramp descending stairs/ramp	accuracy	97.8% - 100%.	Lavenberg - Marquardt algo-rithm with Bayesian regular-ization	IMU (heel) foot switches	5 healthy	Powered ankle foot-orthosis with variable ankle actuation during swing
Wang et al [100]	SMV (a) BPNN (b) RBFNN (c)	6	standing level walking stair ascent stair descent slope ascent slope descent	accuracy	93.3% (b) 91.2% (c)	Trainlm for (a) and (b)	IMUs (right and left thighs and calf, waist) FSRs	5 healthy	-
Villa-Parra [101]	SVM	6	Standing up Sitting Flexion-extension of knee Level walking Resting while standing up Resting while sitting down	accuracy	76%-83% (lower limb muscles) 71%-77% (trunk muscles)	-	sEMG (biceps femoris. erector spinae, vastus lateralis, semitendi-nosus, rectus femoris, gas-trocnemius)	10 healthy	Robotic knee ex-oskeleton for loco-motion mode assistance based on admit-tance control
Hua et al [87]	DNN	6	standing/transition level walking stair ascent stair descent ramp ascent ramp descent	accuracy	0.997	Genetic Algo-rithm Particle Swarm Opti-mization	absolute autoencoder (for joint angles) IMU (back) GRF connecting rod sensors		Heavy load lifting
Goh et al [76]	SSRL (a) SVM-FS (b) SVM-PCA (c) RF-FS (d)	4	walking without exoskeleton walking with exoskeleton at: Zero-torque Low assistive torque High assistive torque	Accuracy (wide spectral frequen-cies) (prominent spectral frequen-cies)	77.8 ± 1.8% (a) 74.3 ± 1.6% (b) 64.3 ± 1.8% (c) 65.9 ± 1.8% (d) 72.9 ± 1.7% (a) 70.2 ± 1.6% (b) 54.8 ± 2.0% (c)	Adam Algo-rithm	EEG (20 channels)	30 healthy (3 ex-cluded)	-

Chinimilli et al [93]	fuzzy inference algorithm	3	level walking uphill walking downhill walking	-	-	-	smart shoes IMUs (thigh and shank) EMG (for testing only) Motion capture	3 healthy	Knee assistive device with automatic virtual impedance modulation
Parri et al [102]	Fuzzy logic	3	level walking Stair ascent Stair descent	accuracy	0.994	-	sensitive insoles (64 optoelectronic sensors) encoders (hip joint angles)	6 healthy	Powered orthosis for hip assistance
Huo et al [95]	Fuzzy logic	5	Level walking Stair ascent Stair descent Slope ascent Slope descent	accuracy latency	97.7% (normal) 97% (abnormal) <32 ± 8.3% of a step	-	Absolute encoder (hip, knee, and ankle) IMUs	4 healthy	Walking assistance using hybrid power assistive control
Novak et al [103]	DT	2	Intention for gait initiation and termination	Median AE (gait initiation/ IMUs/ within-subject trials) Accuracy (gait termination/ IMUs/ within-subject trials)	∼0.08 s (onset) <0.05 s (toe-off) >80%	-	IMUs (thigh, shank, upper arm, foot) foot pressure sensors	10 healthy	Assist to initiate or terminate a step based on intention
Zhang et al [26]	MKL	4	Intention for: Stopping Walking Turning left Turning right	accuracy	74.5% (healthy) 68.4% (SCI)	-	EEG (64 channels placed following 10-20 international system)	1 healthy 1 SCI	Restoration of movement for patients with motor disabilities and induction of cortical plasticity
Gui et al [27]	LDA	4	Intention for: Stopping Level walking Acceleration Deceleration	Recognition rate of steady-state (ROS) Duration of transient state (DOT)	92.40% 1.7 seconds	-	EEG	6 healthy	Admittance control exoskeleton for rehabilitation and motor recovery for paraplegics

Lopez-Larraz et al [28]	SDA	1	Intention for gait initiation	accuracy	88.44 ± 14.56% (healthy) 77.61 ± 14.72% (SCI)	-	EEG (32 channels)	3 healthy 4 SCI	Assist-as-needed exoskeleton for rehabilitation and motor recovery of patients with paralysis
Zheng et al [104]	CCA	3	Intention for motion patterns: Standing Level walking Squatting	Accuracy	>90% (EEG – offline) >90% (multi-modal – online)	-	EEG Sensor for joint position Sensor for foot pressure	4 healthy	-

B.3 Gait Kinetics (Joint Torque/Moments)

Table 3: Existing literature on torque and moment

Authors	Model	Parameter	Performance Metric	Value of Performance Metric	Sensors	Sample	Purpose
Ma et al [86]	KRLS (a) MLPNN (b)	hip joint assist torque	MSE (ranges for five trials)	-32.77 to -28.05 (a) -13.41 to -11.18 (b)	goniometer (hip and knee joint)	10 healthy	Walking assistance for paraplegics + rehabilitation of stroke patients
Gui et al [105]	RBFNN	active torque of hip and knee	RMSE (experimental session 1-3) correlation coefficient r	2.0 N.m r >0.8	EMG (quadriceps femoris and bicep femoris) motion states torque sensors (for validation only)	4 healthy	Assist as needed exoskeleton for rehabilitation training for patients with lower-limb paralysis
Xiong et al [106]	NN	internal joint moment	NRMSE	<7.89%	EMG Joint angles	8 healthy	Exoskeleton for motor rehabilitation
Xiong et al [107]	ELM	Joint moment	VAF	-	EMG (10 muscles)	10 healthy (online database)	Exoskeleton control

B.4 Gait Kinematics (Joint Angles)

Table 4: Existing literature on joint angles

Authors	Model	Parameter	Performance Metric	Value of Performance Metric	Optimization Algorithm	Sensors	Sample	Purpose
Kutilek et al [108]	NN	Joint angles	-	-	Backpropagation	Motion capture	10 healthy	Exoskeleton for rehabilitation

Boudali et al [116]	LRNN (a) LS (b)	hip and knee joint trajectories	RMS	1.36º (hip joint) (a) 2.48º (knee joint) (a)	Levenberg-Marquardt back-propagation (a)	Motion capture	9 healthy	Rehabilitation exoskeleton that assists with locomotion tasks (involves the use of cane)
Mazumder et al [110]	RBFNN	joint trajectories	Tracking error	∼2º (hip) 8-10º (knee)	-	IMUs (thigh, shank, torso, foot) EMG sensors (vastus medialis, medialis hamstring, vastus laterialis, rectus femoris - for gait phases) Foot pressure sensor (for gait phases)	5 healthy	Exoskeleton with intention-based adaptive trajectory control
Lee et al [111]	RBFNN and MLPNN	Joint angles of healthy and pathological leg	accuracy absolute error rate	97.5% 0.25º	error backpropagation algorithm (for MLPNN)	EMG (recuts femoris) tilt sensor (shin center)	-	-
Xie et al [112]	GS-GRNN (a) BPNN (b)	joint angles	Prediction time RMSE	2.38s (a) 16.029s (b) 0.7353 hip (a) 2.6998 knee (a) 3.8373 ankle (a)	Golden selection algorithm (a) Levenberg-Marquardt algorithm (b)	EMG (rectus femoris, semitendinosus, biceps femoris) Joint angle sensor plantar pressure sensor	6 healthy	-
Wang et al [113]	Elam NN (a) BPNN (b) GRNN (c) LSSVM (d)	Joint angles	RMSE (low speed without load conditions)	4.0854 (a) 8.9152 (b) 8.7564 (c) 8.6982 (d)	Backpropagation (a)	EMG (biceps femoris gastrocnemius, rectus femoral, vastus medialis, semitendinosus) Codamotion (for evaluation only)	12 healthy	-
Gomes et al [114]	MLPNN	Joint trajectories (position, velocity and acceleration)	-	(Refer to Figure 14 in [105])	-	-	1 healthy	Active orthosis that adapts gait trajectories based on walking speed

Wu et al [115]	AENN	Hip and knee joint trajectories	Individualized gait pattern generation Correlation coefficient (r) mean absolute deviation (MAD)	0.99 (right hip) 0.97 (right knee) 0.98 (left hip) 0.98 (left knee) 2.60 (right hip) 3.40 (right knee) 2.62 (left hip) 3.64 (left knee)	-	Motion Capture (Noitom TM)	33 healthy	Individual trajectory generation for sharing lower limb exoskeleton (SLEX)
Vallery et al [108]	PCA-CLME BLUE-CLME	Joint trajectories	-	-	-	Torque & angle trajectories (hip and knee joint) EMG (tibialis anterior, gastrocnemius, bicep femoris, rectus femoris -for evaluation)	9 healthy	-
Tanghe et al [97]	PPCA	Hip, knee and ankle joint trajectories	Mean Error	Close to zero (refer to Table III in [97])	-	Vicon cameras split-belt treadmill for GRF (for segmentation)	28 healthy without exo + 5 healthy with exo	feedforward to exoskeleton control algorithms, as compensation to control time delays
Hassan et al [118]	PCA	Joint trajectories of the affected limb	-	-	-	IMUs (thigh, shank and cane) Motion capture system (for verification)	5 patients with hemiparesis	-