

Article

Implementing Gait Kinematic Trajectory Forecasting Models on an Embedded System

Madina Shayne ¹, Leonardo A. Molina ², Bin Hu ³ and Taylor Chomiak ^{2,3,*}

¹ Department of Mechanical and Manufacturing Engineering, University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4, Canada

² CSM Optogenetics Platform, University of Calgary, 3330 Hospital Drive, Calgary, AB T2N 4N1, Canada; leonardo.molina@ucalgary.ca

³ Division of Translational Neuroscience, Department of Clinical Neurosciences, Hotchkiss Brain Institute, Alberta Children's Hospital Research Institute, Cumming School of Medicine, University of Calgary, 3330 Hospital Drive NW, Calgary, AB T2N 4N1, Canada; hub@ucalgary.ca

* Correspondence: tgchomia@ucalgary.ca

Abstract: Smart algorithms for gait kinematic motion prediction in wearable assistive devices including prostheses, bionics, and exoskeletons can ensure safer and more effective device functionality. Although embedded systems can support the use of smart algorithms, there are important limitations associated with computational load. This poses a tangible barrier for models with increased complexity that demand substantial computational resources for superior performance. Forecasting through Recurrent Topology (FReT) represents a computationally lightweight time-series data forecasting algorithm with the ability to update and adapt to the input data structure that can predict complex dynamics. Here, we deployed FReT on an embedded system and evaluated its accuracy, computational time, and precision to forecast gait kinematics from lower-limb motion sensor data from fifteen subjects. FReT was compared to pretrained hyperparameter-optimized NNET and deep-NNET (D-NNET) model architectures, both with static model weight parameters and iteratively updated model weight parameters to enable adaptability to evolving data structures. We found that FReT was not only more accurate than all the network models, reducing the normalized root-mean-square error by almost half on average, but that it also provided the best balance between accuracy, computational time, and precision when considering the combination of these performance variables. The proposed FReT framework on an embedded system, with its improved performance, represents an important step towards the development of new sensor-aided technologies for assistive ambulatory devices.

Keywords: gait; sensor; wearable; embedded system; forecast



Citation: Shayne, M.; Molina, L.A.; Hu, B.; Chomiak, T. Implementing Gait Kinematic Trajectory Forecasting Models on an Embedded System. *Sensors* **2024**, *24*, 2649. <https://doi.org/10.3390/s24082649>

Academic Editors: Toshiyo Tamura and Miguel Velhote Correia

Received: 20 February 2024

Revised: 21 March 2024

Accepted: 17 April 2024

Published: 21 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The advancement in sensor-aided gait kinematic forecasting methods that can incorporate ongoing user-specific sensor signal information on an embedded system could have a significant impact on the design of intelligent lower-limb assistive devices [1]. Unlike fixed-motion capture systems, embedded systems offer a low-cost approach for the collection of high-quality, multi-source data for motion control purposes [2,3]. Moreover, the portability of wearable embedded systems has numerous practical applications for human–robot interactions to enable more efficient integration of a user's natural body motion with an assistive device [4,5]. For instance, the deployment of sensor-aided gait kinematic forecasting technologies on embedded systems can be used to help reduce damage to joints, enhance rehabilitation outcomes, and even reduce the healthcare economic burden by lowering total healthcare system utilization [4,5]. While embedded systems can support the use of gait kinematic forecasting models for a variety of integrated robotic technologies, there are important limitations related to computational load when using complex state-of-the-art models on embedded systems [1]. To address this, the primary

objective of this study was to deploy and evaluate a new computationally lightweight prediction algorithm with reduced complexity and demand for substantial computational resources to enable more effective gait kinematic forecasting on an embedded system. In the following, we will first outline work specifically related to this topic and then describe the methodology and experimental results that are discussed in relation to the use of this new algorithm for sensor-aided gait kinematic forecasting on embedded systems.

2. Related Work

Walking is an essential activity in daily life. Mobility augmentation through the use of wearable assistive ambulatory devices can provide vertical support, assist in lower-limb motion, and improve the quality of life for users [4,6–15]. In addition to the medical applications of wearable assistive ambulatory devices, the impact of various non-medical applications of these devices is also being realized, as they may be able to help healthy individuals perform important activities in daily life [4,5,16]. While there are intricate complexities related to safety regulatory requirements, user acceptance, as well as device reliability and adaptability that need to be considered, assistive ambulatory devices have the potential to help in such situations and may even reduce the burden on healthcare resources [4]. Nevertheless, given the importance of mobility assistance for both medical and non-medical end-user applications, there has been increasing interest in designing more effective and intelligent assistive ambulatory technologies [1,4–6,8,9,11,12,17–29].

Walking trajectory-tracking controllers, characterized by the sensor-aided forecasting of gait kinematic trajectories, represent the next generation in the design of assistive ambulatory devices. This technology aims to achieve natural, stable, and accurate interactive control with respect to human motion intention [3]. It can be used to reduce injury to users as well as compensate for delays in the response time of more complex control systems [1,3,17,18,22,23,30]. Forecasting gait kinematics can also enable targeted functional electrical stimulation at specific points in the gait cycle, leading to more effective rehabilitation therapy, and may even minimize the risk of falling by detecting deviations in the anticipated gait trajectory [1,23,28,29]. However, while lower-limb kinematic trajectory prediction can be used to solve numerous problems facing lower-limb robotics, a significant limiting factor for the implementation of accurate gait forecasting in the design of smart devices is the inability of most forecasting models to support continuous model learning [1]. Continuous learning (CL) would enable adaptive model building to continuously incorporate user-specific [25,31] and current dynamic signal information. This can lead to reliable predictions needed for improved device functionality and user safety [8,10,17,24] while avoiding prediction errors when pretrained models are used under unexpected conditions or conditions that were not included in the initial training process [32–34]. This is particularly relevant for gait, which is dynamically modulated to adjust for differing environmental conditions and meet the needs of ever-changing motor demands [26].

An obvious prerequisite for adaptive model building is the ability for CL to be completed quickly. However, modern prediction models are inherently complex, often requiring hyperparameter optimization and tuning in high-dimensional parameter space [1,6,35–38]. Increasing the number of parameter weights, which is inextricably linked with model architecture complexity, also increases computation time [39]. While increasing model complexity can improve model performance, it is at the expense of rapid CL capabilities [1]. In other words, the fast arrival of new data, together with increasingly slow retraining and optimization processes that are constrained on embedded ambulatory systems, poses a significant barrier to CL. Thus, new methods that permit CL capabilities on embedded systems could have a broad range of practical applications.

Local topological recurrence analysis represents a new computational approach we developed for identifying emergent recurring patterns in a signal's surface topology [40]. Recently, we have expanded on the theoretical and practical properties of local topological recurrence analysis to provide a simple solution for time-series forecasting [38]. Forecasting through Recurrent Topology (FRiT) identifies recurring patterns in local signal shapes to

reveal unique memory traces embedded in a time-series signal that can be used to forecast a system's upcoming time evolution [38]. Furthermore, unlike many other types of models, there is no need for computationally costly optimization and tuning procedures [38]. FReT's simplicity may therefore enable new capabilities for CL to incorporate current dynamic signal information into forecasting. Hence, the objective of this study was to deploy and evaluate FReT-enabled gait kinematic forecasting on an embedded system for prototyping on affordable, embedded hardware that can simplify the integration of sensor-aided gait forecasting into real-world applications.

3. Materials and Methods

3.1. Artificial Neural Network Models

Artificial neural networks are a type of artificial intelligence technology that mimics the human brain's powerful ability to recognize patterns [39]. These models have been used successfully for modelling a broad range of time-series data [36,38,41,42]. The task of the artificial neural network is to model the underlying data-generating process during training so that valid forecasts can be made when the parameterized model is subsequently presented with new input data [35]. The most widely used and often preferred models when building artificial neural network forecasting models are those with a Multilayer Perceptron architecture, given its computational efficiency and efficacy and its ability to be extended to deep learning (Figure 1) [1,35,36,38,41,43,44]. Mathematically, a basic artificial neural network model (NNET) can be represented as follows:

$$x_{t+s} = \beta_0 + \sum_{j=1}^D \beta_j g\left(\gamma_{0j} + \sum_{i=1}^m \gamma_{ij} x_{t-(i-1)d}\right) \quad (1)$$

in which there are two critical hyperparameters that need to be chosen, the embedding dimension, m , which captures the autocorrelation structure of the time series, and the number of hidden units, D [35,41,43]. For an artificial neural network model with a deep architecture (D-NNET), the number of hidden layers also needs to be determined [38,45].

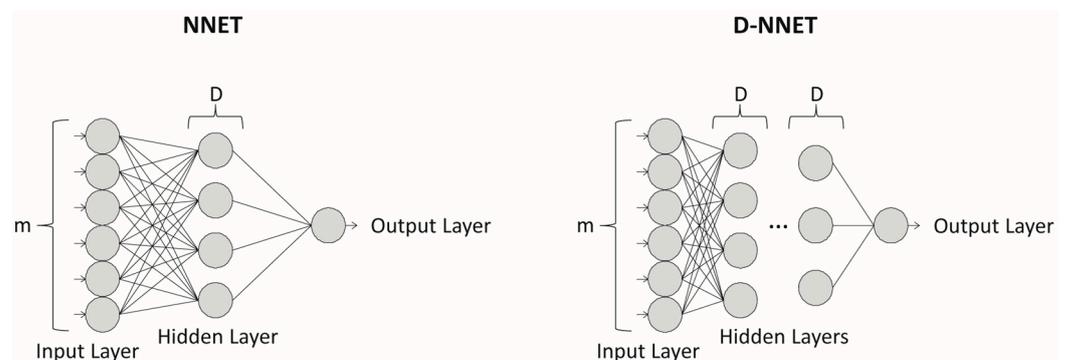


Figure 1. Basic artificial neural network architecture for time-series forecasting. The widely used NNET model (left), given its versatility in modelling a wide range of time-series data, and its ability to be extended to deep learning (right). The two critical hyperparameters that need to be chosen correctly for NNET models are the embedding dimension, m , and the number of hidden units, D . For NNET models with a deep architecture (D-NNET), the number of hidden layers also needs to be determined.

Given that there is no general rule that can be followed to select the most appropriate hyperparameters, training data are used to estimate the optimal hyperparameter values and model architectures based on minimizing error-related terms during training. This step is typically computationally expensive. Using this approach, pretrained hyperparameter-optimized NNET models can be built. These models represent a simplified approach adept at handling tasks with constrained data availability, owing to their streamlined architecture and efficient training process. Yet, their simplicity can sometimes come at

the expense of performance, as they are not able to extract deep features [46]. Conversely, D-NNET architectures describe models that utilize multiple hidden layers to represent features at higher and more abstract levels that are learned from the data [46]. D-NNET architectures, characterized by their increased complexity and capacity for performance, often demand substantial computational resources and extensive datasets for more effective training and deployment. However, the greater the number of weight parameters relative to the size of the training data, the greater the ability of the network to memorize the idiosyncrasies of individual observations. As a result, model generalization can be lost, leading to the development of a model that can be of little use in forecasting [39]. While the flexibility of artificial neural network models provides a potentially powerful forecasting tool, hyperparameter/parameter determination can complicate the design process [39].

3.2. Forecasting through Recurrent Topology

Forecasting through Recurrent Topology (FReT) is a time-series data prediction algorithm method that is based on learning recurrent patterns in a series' local topology [38]. It is a versatile algorithm that reduces computational complexity and cost, with demonstrated feasibility using a variety of dynamic systems [38]. Unlike highly parameterized models, there is no need for computationally costly hyperparameter optimization and tuning procedures in high-dimensional parameter space.

FReT works by taking a data sequence where \vec{x} represents a one-dimensional time-series vector:

$$\vec{x} = (x_1, x_2, x_3, \dots, x_n) \quad (2)$$

and generates a Euclidean distance matrix (D), which is remapped to a local 3×3 neighbourhood topological matrix (T'):

$$\left(T'_{ij}\right)_8 : D \rightarrow T' \quad (3)$$

$$\left(T'_{ij}\right)_8 = \sum_{q=1}^8 s(g_q - g_0)2^{q-1}; s(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (4)$$

A binary code is then created by moving around the central point, g_0 , where a single integer value is calculated based on the sum of the binary code elements (0 or 1) multiplied by the eight positional weights (Figure 2a). This represents 8-bit binary coding, where there are 2^8 (256) different possible integer values that are partitioned into sextiles (six flattened layers) [38], generating a two-dimensional local topological matrix (T') composed of a set of six integer values, $\mathbb{Z} = \{1, 2, 3, 4, 5, 6\}$ (Figure 2b–d). Each point along the two-dimensional matrix diagonal represents a point in the signal sequence and its associated row vector (Figure 2d).

From T' , element-wise differences in all prior row vectors are computed with respect to the last row vector (an index of the system's current state), generating a 1 (true) if their difference equals zero, otherwise 0 (false). These binary elements are summed and divided by the length of the row vector. This generates a similarity metric (Figure 3a) that ranges from 0 to 1 that can be used to find the index (+3 to account for the padded local topological matrix and beginning one step in the future) of the encoded training data topological archetype(s). This similarity metric differentially weights the importance of each part of the input training data. This produces a one-dimensional weight vector with respect to the system's current state. The higher the values, the more closely the topological sequences align with the system's current state based on topological patterning [38]. These topological archetypes ultimately reveal unique memory traces of past system behaviour(s) that can be used to construct a single multi-step-ahead embodied model of a dynamic system's upcoming unseen dynamics [38] (Figure 3b,c). This approach has recently shown promise for forecasting human gait kinematics [38].

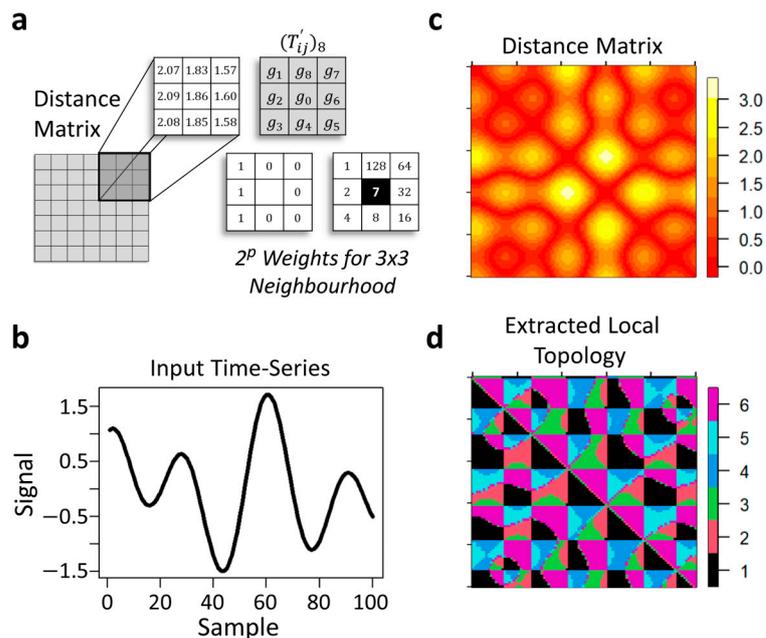


Figure 2. Extracting local topological patterning. A schematic illustrating the basic premise behind local topological pattern extraction (a). The process starts by creating a distance matrix from an input time-series signal. A binary code is then created by moving around the central point, g_0 , where a single integer value is calculated based on the sum of the binary code elements (0 or 1) multiplied by the eight positional weights. This is carried out over the entire distance matrix. (b–d) present an example of this process. A simulated time series (b), its traditional distance matrix (c), and the extracted local topological matrix (d). See text for additional details.

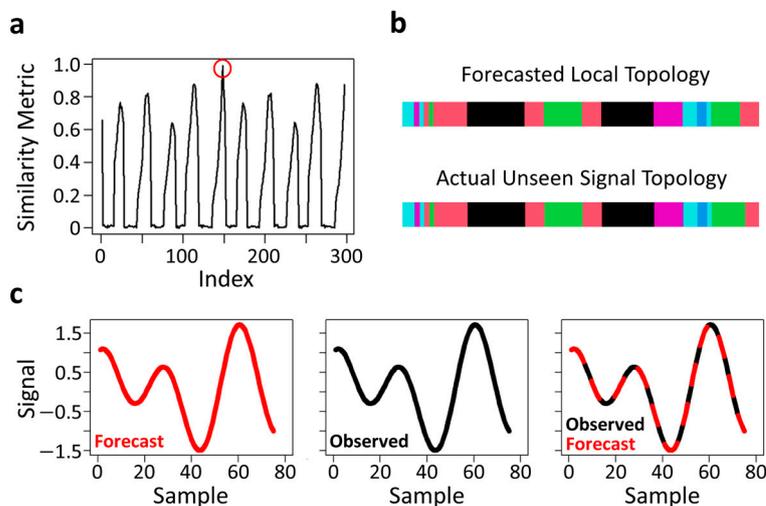


Figure 3. Topological patterning and archetype identification. A similarity metric related to the index of the training data that is used for topological archetype identification that effectively reduces to a simple maximization problem ((a), red circle). Panels b–c present the forecasted local topology of this identified archetype and the actual unseen signal topology of a simulated time-series signal ((b); very similar although not quite identical) that is used to construct an embodied model of the system’s upcoming dynamics (c). The colors in panel b reflect the set of six integer values.

3.3. Kinematic Trajectory Forecasting

An advantage of FReT is that its forecasts are based on the original data and associated scale [38]. Kinematic trajectory forecasts based on FReT can therefore be generated directly from the input window data. For NNET and D-NNET models, the training dataset was first used to estimate hyperparameters and model architectures before being deployed to the

embedded system (Figure 4a,b). For testing, data were z-score-scaled to standardize model training, and the data of each input window were first z-score-scaled before being fed into the model (Figure 4c). The forecasted gait data, a z-score output, were then converted back to the original sensor scale based on the attributes (mean and standard deviation) of the z-scored input window data. This enabled comparison to the true sensor signal test data (Figure 4c).

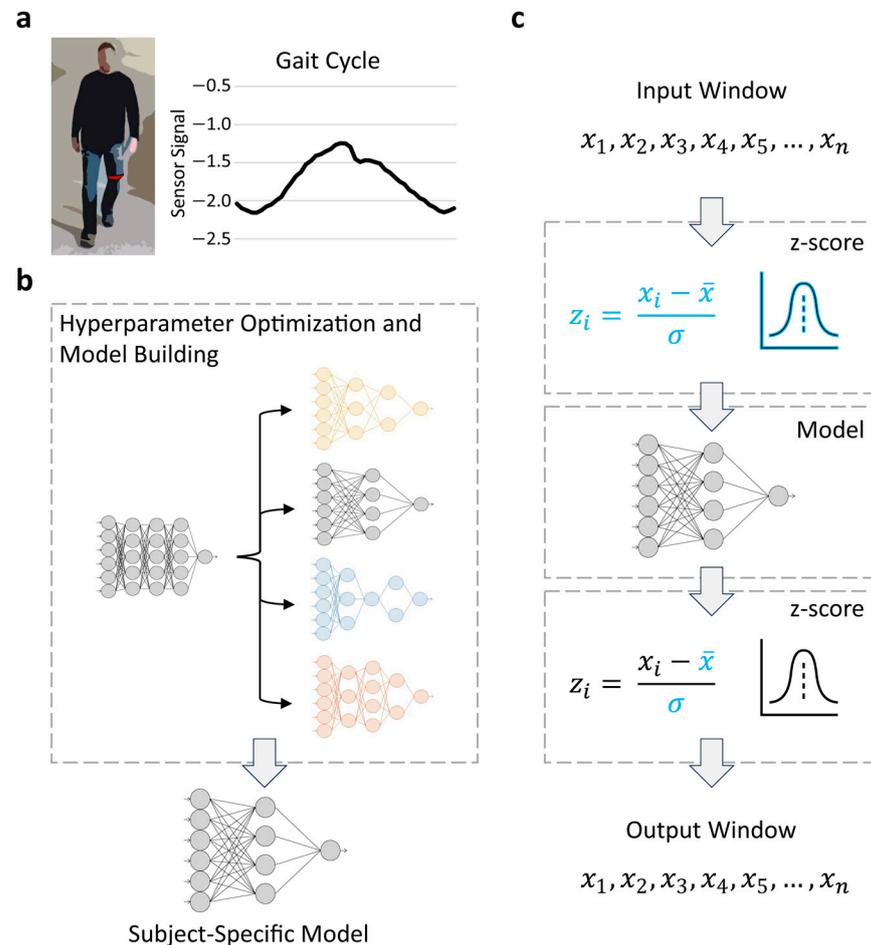


Figure 4. A schematic of the data processing workflow. Gait data were collected by a wearable sensor (illustrated in red in (a), left). The sensor is attached to the leg just above the patellofemoral joint line for recording gait kinematics based on hip flexion/extension using the Ambulosono system ((a), right). A subject’s training dataset was used to estimate hyperparameters and model architectures specific to an individual (subject-specific model) for NNET and D-NNET models (b). This was done on an external computer. For testing on the embedded system, the data for each subject’s input window were first z-score-scaled before being fed into their subject-specific model (c). The forecasted gait data, with the output as a z-score, were then converted back to the original sensor scale based on the attributes (mean and standard deviation) of the z-scored input window data. This allowed comparison to the true sensor signal test data (c).

3.4. Gait Sensor

Gait data were analyzed from a heterogeneous sample of fifteen ($n = 15$) adult participants without neurological abnormalities previously collected using the first-generation Ambulosono wearable sensor system [47]. The sensor system is based on using motion processor data consisting of a 3-axis Micro-Electro-Mechanical System (MEMS)-based gyroscope and a 3-axis accelerometer. The system and its firmware continuously record gait cycle dynamics and gait metrics while controlling for angular excursion and drift [47–50]. The sensor is attached to the leg just above the patellofemoral joint line through the use

of a high-performance thigh band. This is the optimal location for recording gait features based on hip flexion/extension using this system [47–50].

3.5. Gait Data

The software application utilizes sensor data for step parameter calculations based on its ability to automatically detect hip flexion/extension in real time. The algorithm works by using a biomechanics model that uses circumference geometry and its concepts of radius and radians with limb length to functionally relate gait parameters to hip flexion/extension angle [47]. For this study, we used hip joint flexion/extension data, which are fundamental to this biomechanical model of gait. Hip joint flexion/extension data are also important for gait rehabilitation and human locomotion assistance [16]. While wearing the sensor, participants were asked to complete a 30 m walk at a self-selected pace [47]. Each participants' time-series data (600 data points) were partitioned (300 data points), with the partitions being roughly split between training and testing sets.

3.6. Embedded System

The embedded system that was prototyped was the Raspberry Pi 4 (<https://www.raspberrypi.com/>) (Figure 5). The Raspberry Pi is a very popular system that is affordable, well supported, and can run multiple code threads simultaneously. It is also suitable for performing multifaceted tasks, such as running complex control systems, and can be easily integrated with next-generation low-cost and low-power-consumption motion processors for future development. The Raspberry Pi, therefore, represents a valid prototyping base and highly versatile system that has the potential to be easily incorporated into lower-limb assistive ambulatory devices. In fact, the Raspberry Pi 4 was recently used as an embedded system to read sensor data and perform real-time exoskeleton control and optimization [26]. The Raspberry Pi 4 consists of a Broadcom BCM2711 Quad core Cortex-A72 (ARM v8) 64-bit SoC @1.5 GHz processor.

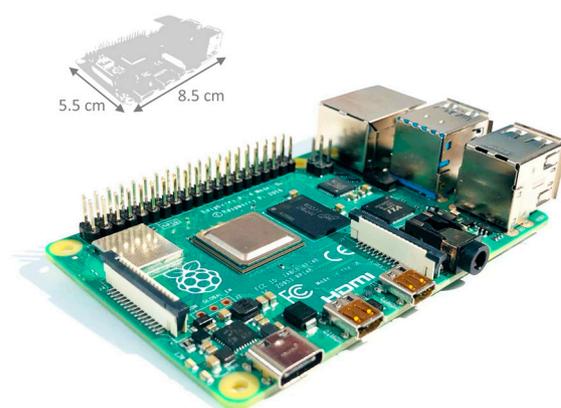


Figure 5. The embedded system that was prototyped. An illustration of the Raspberry Pi 4 and its dimensions (inset).

3.7. Model Deployment

Forecast model deployment was accomplished with R (<https://www.R-project.org/>) (accessed on 24 October 2023) by first installing Visual Studio Code on the Raspberry Pi, a popular source-code editor that can be used with a variety of programming languages, followed by the R extension. For NNET model implementation, the tsDyn package [43] was deployed. For D-NNET model implementation, the nnfor package [45] was deployed. FRET does not require any R user library packages.

3.8. Data Analysis

The models were evaluated by feeding the gait test set data into the embedded system, and the time to execute the forecast and the accuracy of the forecast were determined.

4. Results

4.1. Forecasting Models

The NNET and D-NNET model architectures were coded in R [43,45] and estimated on an external computing source. The model hyperparameters were determined first via grid search to estimate an optimal value of hidden units, embedding dimension, and number of hidden layers (for D-NNET), similar to that carried out previously [1,38]. Once the model architectures were selected for each individual, these subject-specific hyperparameter-optimized NNET and D-NNET model architectures were deployed to the embedded system. FReT, which does not require hyperparameter optimization, was directly deployed to the embedded system for comparison with these pretrained hyperparameter-optimized NNET and D-NNET models.

The NNET and D-NNET models were each evaluated under two conditions. The first condition represents a static model. Here, the model weights are fixed, and each sliding input window was simply fed into the pretrained model for predicting the unseen output window data (Figure 7a,c). For the second condition, the updated NNET and D-NNET models (Figure 7b,d), the input window data were first used to retrain the model weights before predicting the unseen output window data. This approach provides a mechanism to update the model with real-time information embedded in the gait data signal that may not be appropriately modelled using the fully pretrained static models. FReT, by design, always updates based on the input window data, where the two-dimensional topological image reflects recurrent patterning in the gait signal's surface topology (Figure 7e). Illustrative examples of forecasted output window kinematic gait trajectories for all the prediction approaches are also shown in Figure 7f–j.

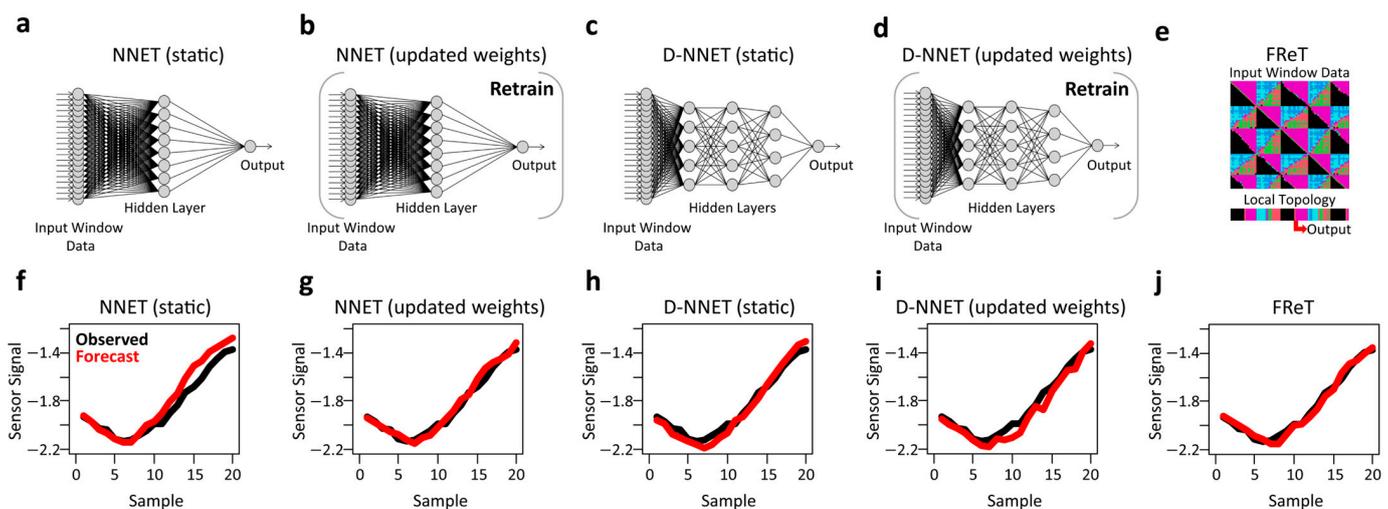


Figure 7. Comparison of models and their outputs. For all NNET and D-NNET models, hyperparameters were determined on an external computing source before being deployed to the Raspberry Pi. NNET and D-NNET models were run under two conditions. The first represents a static model where the model weights are fixed, and each sliding input window was simply fed into the fully pretrained model for predicting the unseen output window data (a,c). For the second condition, the model weights are updated by retaining with the input window data (b,d). Panel e illustrates topological archetype identification based on a simple maximization problem. (f–j) present example forecasted data with an input window of 80 data points and an output (forecast) window of 20 data points for each of the corresponding models shown above (a–e). Ground truth is shown in black, while each model's predicted trajectory is shown in red.

4.2. Model Accuracy, Intra-Individual Variability, and Computational Time

Using the models presented in Figure 7a–e, we summarized the accuracy results across all the forecasting approaches for all fifteen participants (Table 1). The mean NRMSE

is presented along with a measure of the mean intra-individual dispersion generated from 10 iterations per participant (see Materials and Methods). There, we can see that, unlike the network models, FReT exhibited greater precision (Table 1). Not only was FReT more consistent than the network models, but it was also more accurate than all the models tested (Table 1), with a NRMSE that was, on average, 46.2% lower across all the models/experiments.

Table 1. Accuracy comparison between models.

| Model | Experiment 1 | | Experiment 2 | |
|------------------|--------------|------------|--------------|------------|
| | NRMSE | Dispersion | NRMSE | Dispersion |
| NNET (static) | 0.237 | 0.047 | 0.157 | 0.027 |
| NNET (updated) | 0.143 | 0.004 | 0.146 | 0.008 |
| D-NNET (static) | 0.203 | 0.055 | 0.188 | 0.054 |
| D-NNET (updated) | 0.535 | 0.023 | 0.221 | 0.014 |
| FReT | 0.115 | 0.000 | 0.097 | 0.000 |

Experiment 1: sliding input window of 50 data points and a prediction output window of 15 data points (approx. 400 ms). Experiment 2: sliding input window of 80 data points and a prediction output window of 20 data points (approx. 550 ms). NRMSE: normalized root-mean-square error. Dispersion represents the standard deviation of the mean NRMSE across 10 iterations.

Next, to ensure that FReT's accuracy was not at the expense of computational cost, the computational time for each model was evaluated and is summarized in Table 2. While the static NNET and D-NNET models offered the best performance in terms of computational time (Table 2), they were not the most accurate (Table 1). The most accurate network model was the updated NNET model (Table 1). However, while the updated NNET model exhibited the best performance of all the artificial neural network models tested in terms of accuracy, FReT had less intra-individual variability, was more accurate, and was faster than the best-performing artificial neural network model (Tables 1 and 2). Moreover, FReT also provided the best balance with respect to accuracy, intra-individual variability, and computational time when considering the combination of these performance variables. This is easily seen when plotted in scaled multidimensional space (Figure 8). There, it is evident that FReT tends to be located on the left, mid-to-lower side of the plot towards the ideal situation (i.e., the left bottom corner, which represents the idealized condition of a model's performance variables approaching zero: zero error, zero computational time, and zero dispersion) (Figure 8). In fact, of all the models, FReT was closest to this idealized condition for both experiment 1 and experiment 2 (Figure 8 and Table 3).

Table 2. Comparison of computational time between models.

| Model | Experiment 1 | Experiment 2 |
|------------------|--------------|--------------|
| | ms | ms |
| NNET (static) | 9.534 | 11.11 |
| NNET (updated) | 195.0 | 218.8 |
| D-NNET (static) | 17.83 | 19.99 |
| D-NNET (updated) | 186.6 | 395.3 |
| FReT | 80.02 | 200.4 |

Experiment 1: sliding input window of 50 data points and a prediction output window of 15 data points (approx. 400 ms). Experiment 2: sliding input window of 80 data points and a prediction output window of 20 data points (approx. 550 ms).

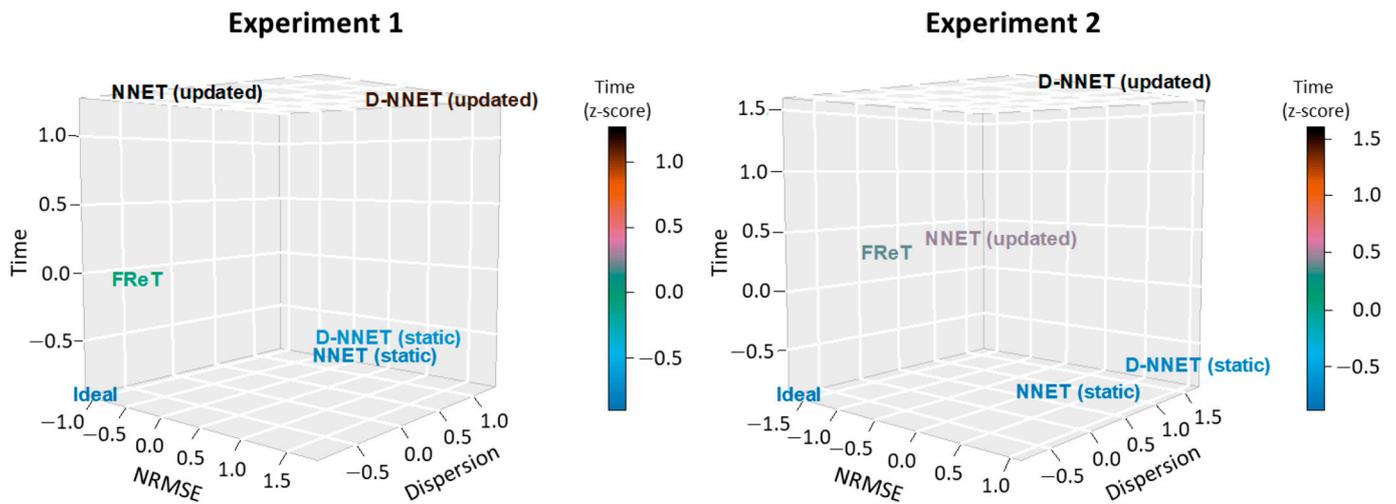


Figure 8. Model performance variables in multidimensional space. Performance variables for each model were z-score-scaled for plotting in multidimensional space. The bottom left corner (“Ideal”) represents the ideal condition of a perfect model with performance variables approaching zero.

Table 3. Distance between models and the idealized condition of a perfect model.

| Model | Experiment 1 | Experiment 2 |
|------------------|--------------------|--------------------|
| | Euclidean Distance | Euclidean Distance |
| NNET (static) | 2.32 | 2.40 |
| NNET (updated) | 2.33 | 2.36 |
| D-NNET (static) | 2.52 | 3.55 |
| D-NNET (updated) | 3.74 | 3.83 |
| FReT | 1.10 | 1.77 |

Euclidean distance between the idealized condition of a perfect model (i.e., zero error, zero intra-individual variability, and zero computational time) and the evaluated forecasting models.

4.3. Model Inter-Individual Variability

Our previous analyses focused on intra-individual model dispersion, which would tend to be more important for real-world applications as subject-specific training is preferable if the gait trajectory is to be used within a wearable device framework [6]. However, we also evaluated the dispersion of the average NRMSE per participant across all the participants to further showcase the utility of FReT. Dispersion in this sense represents inter-individual variability and can provide information regarding a model’s generalizability. The NRMSE distribution densities for all fifteen participants are shown for each model in Figure 9. There, it can be seen that FReT exhibited the highest density of all the models, with the peak of the light-tailed distribution density centered around the average NRMSE per participant (Figure 9). Together, these data, in addition to the accuracy, intra-individual variability, and computational time data, lend support to the idea that patterns in local topological recurrences embedded in wearable sensor data can be used to effectively forecast human gait kinematics.

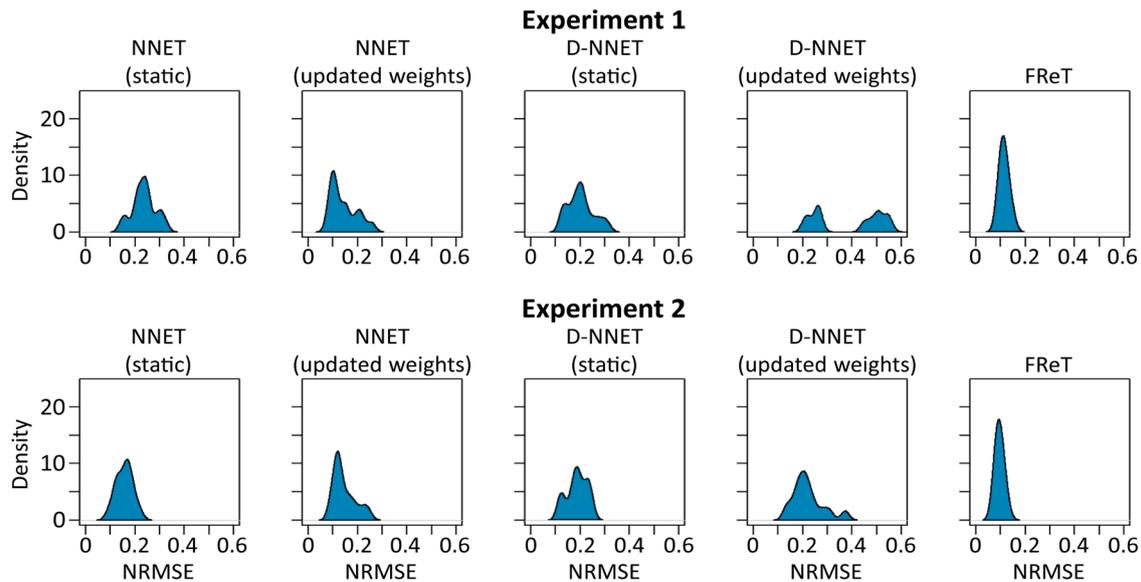


Figure 9. Distribution of NRMSE for FReT and the artificial neural network models. For each experiment, the distribution density of the average NRMSE per participant for all fifteen participants is shown for each model. The updated NNET model was the best-performing network model, while FReT was the best-performing model overall.

5. Discussion

5.1. Complex Prediction Models

In recent years, artificial neural network models have attracted increasing attention with respect to time-series forecasting. It is well established that these models represent a versatile computational framework that can be used for modelling a broad range of time-series data [35,36,38,41,43], including gait trajectory prediction within the context of machine learning-enabled assistive ambulatory device design [1]. For example, using wearable sensor data, Su et al. proposed a gait trajectory prediction model that was based on a long short-term memory (LSTM) network with a weighted discount loss function [6]. This approach was, overall, able to predict the gait trajectory for multiple time frames, although the model was designed to more accurately predict the gait trajectory of the immediate future (e.g., 100 ms) [6]. Karakish et al. noted the limitations associated with computational load and relative inference time with the use of more complex models such as LSTM on embedded systems [1]. They decided to explore the use of simpler artificial neural network models, including D-NNET and convolutional neural network (CNN) models, on an embedded system to predict motion sensor data. Using these models, they were able to achieve comparable results, even after reducing the size of the networks and with forecast horizons of 200 ms [1]. However, it is important to point out that unlike FReT, all of these models still rely on neural network-based architectures and model hyperparameters that need to be chosen appropriately [1,6,35,36,38,41,43]. With these and many other types of complex forecasting models, there is no general rule that can be followed to select the most appropriate values. These values, therefore, need to be estimated based on available training data [41,43]. In fact, it has already been pointed out in the literature that this learning process, particularly for highly complex prediction models, needs to be improved, as it can take extended periods of time which may be unacceptable for time-series prediction in real-world applications [46]. Forecasting gait with FReT, on the other hand, avoids the need for the selection of hyperparameters and the uncertainty associated with their selection. In this study, we have also shown that FReT may offer a simple approach to predict gait kinematic trajectory at horizons > 200 ms. This study further highlights that more complex models are not always necessary for top predictive

performance and can impose unnecessary computational load and power consumption constraints [38].

5.2. Embedded Systems

Within the framework of developing an embedded active assistive ambulatory system, it has already been suggested that it makes more sense to develop a system that uses wearable sensors for both the training and deployment stages rather than relying on a fixed-image capture system to record gait motion [1]. While models have been built based on fixed-motion capture systems (e.g., [37]), the uncertainty related to their transferability outside a fixed-motion capture environment has been less well studied. Moreover, problems can arise in situations where the motion capture markers used for gait are obscured by arm swinging or hand rails that may be needed for safety in certain clinical populations [51]. Unlike fixed-motion capture systems, embedded systems have numerous practical and important applications in many areas of rehabilitation science and biomedical engineering [1,6,9,17–25,28,29]. For example, as previously noted [37], the accurate sensor-aided prediction of gait kinematic trajectories can serve as a feedforward mechanism to powered devices instead of predominantly relying on feedback sensors, effectively serving to improve device performance by avoiding alterations in the user's natural gait trajectories [37]. The implementation of FReT on embedded systems for sensor-aided gait kinematic trajectory prediction may also have several important operational advantages. First, being able to accurately predict gait time-series data while avoiding the drawbacks associated with modern prediction models, including random matrices and sensitivity to input hyperparameter selection, may help facilitate the design of a control system that works stably under a variety of conditions. Second, given that FReT does not require optimization and tuning techniques, it can be easily adjusted based on natural variations in walking speed. This is particularly important as it eliminates the need for pretraining data collection that is needed for commonly used machine learning models for model building. The requirement for collecting training data places an increased burden on both participants' and clinicians' time and is subject to several technical factors that may impact the quality of training data generalization. Finally, FReT does not require the time-consuming grid searches that are needed by other models during model building [1,6]. This can enable an adaptive, user-centric forecasting approach within an embedded framework that utilizes the fast arrival of new data for predicting upcoming gait dynamics.

5.3. Study Limitations

We acknowledge that there are several study limitations. First, the gait data for this study were collected at an individual's preferred, self-selected walking speed. Data for variable walking speed were not evaluated, which is likely to reflect more dynamic, naturalistic environments. However, this may pose much less of a problem for FReT compared to many other types of complex forecasting models. In fact, several studies have found that commonly used prediction models for measures of gait perform much poorer when not trained on intra-subject data [1,6,34]. This can be problematic for traditional models requiring model building as the training data would need to reflect all potential environments, for all individuals, that may be encountered during deployment. FReT effectively avoids this issue as it was initially designed to enable CL-based forecasting on prior signal dynamics. Nevertheless, whether FReT can perform as well under dynamic walking conditions needs further testing. Second, this study also did not focus on clinical populations with variations in gait function, including individuals with walking disabilities and/or a history of falls. These would need to be evaluated to determine the applicability of this approach in a wider context and to understand its potential application to fall prevention. Finally, we did not evaluate the integration of multi-source information for motion intention [2,3]. However, FReT can be used for decoding multidimensional systems and predicting multidimensional signal topology [38].

6. Conclusions

The overall goal of this study was to deploy FReT on an embedded system to evaluate its ability to forecast wearable lower-limb motion sensor data. FReT was compared with several different artificial neural network and deep artificial neural network model architectures. This included models with both static model weight parameters and iteratively updated model weight parameters, with the latter serving as a mechanism to update the models with real-time information embedded in the gait signal that may not be appropriately modelled using fully pretrained static models. Our results indicate that FReT provided the best balance with respect to performance, more closely matching the idealized condition of a perfect model than all the other models evaluated. The proposed FReT framework has the advantage of being more accurate with a lower computational load and better consistency relative to the best-performing NNET model for predicting gait kinematic trajectories. Taken together, this study suggests that FReT can be used on an embedded system to effectively forecast gait cycle kinematics, supporting a new continuous learning capability that may work stably across a variety of conditions.

Author Contributions: Conceptualization, T.C.; investigation, M.S.; methodology, T.C., M.S. and L.A.M.; analysis, M.S. and T.C.; resources, L.A.M. and B.H.; supervision, T.C.; visualization, T.C. and M.S.; data curation, B.H.; writing—original draft preparation, T.C. and M.S.; writing—review and editing, T.C., M.S., L.A.M. and B.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The human data used in this study were previously collected with informed consent and ethics approval by the ethics committee of the University of Calgary University Ethics Board for Human Research (REB13-0009) [47].

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study as noted above.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the Hotchkiss Brain Institute and the CSM Optogenetics Platform for their continued support.

Conflicts of Interest: B.H. is the founder of Ambuloso International Development Inc., a wearable sensor startup consulting firm. All the other authors declare no competing interests.

References

1. Karakish, M.; Fouz, M.A.; ELsawaf, A. Gait Trajectory Prediction on an Embedded Microcontroller Using Deep Learning. *Sensors* **2022**, *22*, 8441. [[CrossRef](#)] [[PubMed](#)]
2. Mobarak, R.; Tigrini, A.; Verdini, F.; Al-Timemy, A.H.; Fioretti, S.; Burattini, L.; Mengarelli, A. A Minimal and Multi-Source Recording Setup for Ankle Joint Kinematics Estimation During Walking Using Only Proximal Information from Lower Limb. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2024**, *32*, 812–821. [[CrossRef](#)] [[PubMed](#)]
3. Zhang, Y.-P.; Cao, G.-Z.; Li, L.-L.; Diao, D.-F. Interactive Control of Lower Limb Exoskeleton Robots: A Review. *IEEE Sens. J.* **2024**, *24*, 5759–5784. [[CrossRef](#)]
4. Rupal, B.S.; Rafique, S.; Singla, A.; Singla, E.; Isaksson, M.; Virk, G.S. Lower-Limb Exoskeletons: Research Trends and Regulatory Guidelines in Medical and Non-Medical Applications. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417743554. [[CrossRef](#)]
5. Young, A.J.; Ferris, D.P. State of the Art and Future Directions for Lower Limb Robotic Exoskeletons. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2017**, *25*, 171–182. [[CrossRef](#)] [[PubMed](#)]
6. Su, B.; Gutierrez-Farewik, E.M. Gait Trajectory and Gait Phase Prediction Based on an LSTM Network. *Sensors* **2020**, *20*, 7127. [[CrossRef](#)] [[PubMed](#)]
7. Tiboni, M.; Borboni, A.; Vèrité, F.; Bregoli, C.; Amici, C. Sensors and Actuation Technologies in Exoskeletons: A Review. *Sensors* **2022**, *22*, 884. [[CrossRef](#)] [[PubMed](#)]
8. Murray, S.; Goldfarb, M. Towards the Use of a Lower Limb Exoskeleton for Locomotion Assistance in Individuals with Neuromuscular Locomotor Deficits. In Proceedings of the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Diego, CA, USA, 28 August–1 September 2012; Volume 2012, p. 1912. [[CrossRef](#)]

9. Kim, M.; Hargrove, L.J. A Gait Phase Prediction Model Trained on Benchmark Datasets for Evaluating a Controller for Prosthetic Legs. *Front. Neurorobot.* **2023**, *16*, 288. [[CrossRef](#)] [[PubMed](#)]
10. Anam, K.; Al-Jumaily, A.A. Active Exoskeleton Control Systems: State of the Art. *Procedia Eng.* **2012**, *41*, 988–994. [[CrossRef](#)]
11. Taborri, J.; Scalona, E.; Rossi, S.; Palermo, E.; Patane, F.; Cappa, P. Real-Time Gait Detection Based on Hidden Markov Model: Is It Possible to Avoid Training Procedure? In Proceedings of the 2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings, Turin, Italy, 7–9 May 2015; pp. 141–145. [[CrossRef](#)]
12. Zhou, J.; Yang, S.; Xue, Q. Lower Limb Rehabilitation Exoskeleton Robot: A Review. *Adv. Mech. Eng.* **2021**, *13*, 16878140211011862. [[CrossRef](#)]
13. Androwis, G.J.; Sandroff, B.M.; Niewrzol, P.; Fakhoury, F.; Wylie, G.R.; Yue, G.; DeLuca, J. A Pilot Randomized Controlled Trial of Robotic Exoskeleton-Assisted Exercise Rehabilitation in Multiple Sclerosis. *Mult. Scler. Relat. Disord.* **2021**, *51*, 102936. [[CrossRef](#)] [[PubMed](#)]
14. van Nes, I.J.W.; van Djijseldonk, R.B.; van Herpen, F.H.M.; Rijken, H.; Geurts, A.C.H.; Keijsers, N.L.W. Improvement of Quality of Life after 2-Month Exoskeleton Training in Patients with Chronic Spinal Cord Injury. *J. Spinal Cord Med.* **2022**. [[CrossRef](#)]
15. Asano, M.; Rushton, P.; Miller, W.C.; Deathe, B.A. Predictors of Quality of Life among Individuals Who Have a Lower Limb Amputation. *Prosthet. Orthot. Int.* **2008**, *32*, 231–243. [[CrossRef](#)]
16. Chen, B.; Ma, H.; Qin, L.Y.; Gao, F.; Chan, K.M.; Law, S.W.; Qin, L.; Liao, W.H. Recent Developments and Challenges of Lower Extremity Exoskeletons. *J. Orthop. Transl.* **2016**, *5*, 26–37. [[CrossRef](#)]
17. Vu, H.T.T.; Dong, D.; Cao, H.L.; Verstraten, T.; Lefeber, D.; Vanderborght, B.; Geeroms, J. A Review of Gait Phase Detection Algorithms for Lower Limb Prostheses. *Sensors* **2020**, *20*, 3972. [[CrossRef](#)]
18. Clemens, S.; Kim, K.J.; Gailey, R.; Kirk-Sanchez, N.; Kristal, A.; Gaunaud, I. Inertial Sensor-Based Measures of Gait Symmetry and Repeatability in People with Unilateral Lower Limb Amputation. *Clin. Biomech.* **2020**, *72*, 102–107. [[CrossRef](#)]
19. Lai, D.T.H.; Begg, R.K.; Palaniswami, M. Computational Intelligence in Gait Research: A Perspective on Current Applications and Future Challenges. *IEEE Trans. Inf. Technol. Biomed.* **2009**, *13*, 687–702. [[CrossRef](#)] [[PubMed](#)]
20. Sawicki, G.S.; Beck, O.N.; Kang, I.; Young, A.J. The Exoskeleton Expansion: Improving Walking and Running Economy. *J. Neuroeng. Rehabil.* **2020**, *17*, 25. [[CrossRef](#)] [[PubMed](#)]
21. Pan, C.T.; Chang, C.C.; Sun, P.Y.; Lee, C.L.; Lin, T.C.; Yen, C.K.; Yang, Y.S. Development of Multi-Axis Motor Control Systems for Lower Limb Robotic Exoskeleton. *J. Med. Biol. Eng.* **2019**, *39*, 752–763. [[CrossRef](#)]
22. Tanghe, K.; De Groote, F.; Lefeber, D.; De Schutter, J.; Aertbelien, E. Gait Trajectory and Event Prediction from State Estimation for Exoskeletons during Gait. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2020**, *28*, 211–220. [[CrossRef](#)]
23. Zaroug, A.; Proud, J.K.; Lai, D.T.H.; Mudie, K.; Billing, D.; Begg, R. Overview of Computational Intelligence (CI) Techniques for Powered Exoskeletons. *Stud. Comput. Intell.* **2019**, *776*, 353–383. [[CrossRef](#)]
24. Torricelli, D.; Cortés, C.; Lete, N.; Bertelsen, Á.; Gonzalez-Vargas, J.E.; Del-Ama, A.J.; Dimbwadyo, I.; Moreno, J.C.; Florez, J.; Pons, J.L. A Subject-Specific Kinematic Model to Predict Human Motion in Exoskeleton-Assisted Gait. *Front. Neurorobot.* **2018**, *12*, 18. [[CrossRef](#)]
25. Wu, X.; Liu, D.X.; Liu, M.; Chen, C.; Guo, H. Individualized Gait Pattern Generation for Sharing Lower Limb Exoskeleton Robot. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 1459–1470. [[CrossRef](#)]
26. Slade, P.; Kochenderfer, M.J.; Delp, S.L.; Collins, S.H. Personalizing Exoskeleton Assistance While Walking in the Real World. *Nature* **2022**, *610*, 277–282. [[CrossRef](#)] [[PubMed](#)]
27. Dobson, A.; Murray, K.; Manolov, N.; Davanzo, J.E. Economic Value of Orthotic and Prosthetic Services among Medicare Beneficiaries: A Claims-Based Retrospective Cohort Study, 2011–2014. *J. Neuroeng. Rehabil.* **2018**, *15*, 55. [[CrossRef](#)]
28. Parthasarathy, A.; Megharjun, V.N.; Talasila, V. Forecasting a Gait Cycle Parameter Region to Enable Optimal FES Triggering. *IFAC-PapersOnLine* **2020**, *53*, 232–239. [[CrossRef](#)]
29. Rahman, H.; Kumbala, A.; Megharjun, V.N.; Talasila, V. Real-Time Heel Strike Parameter Estimation for FES Triggering. *Lect. Notes Electr. Eng.* **2022**, *903*, 749–760. [[CrossRef](#)]
30. Zaroug, A.; Lai, D.T.H.; Mudie, K.; Begg, R. Lower Limb Kinematics Trajectory Prediction Using Long Short-Term Memory Neural Networks. *Front. Bioeng. Biotechnol.* **2020**, *8*, 362. [[CrossRef](#)]
31. Kale, A.; Sundaresan, A.; Rajagopalan, A.N.; Cuntoor, N.P.; Roy-Chowdhury, A.K.; Krüger, V.; Chellappa, R. Identification of Humans Using Gait. *IEEE Trans. Image Process.* **2004**, *13*, 1163–1173. [[CrossRef](#)]
32. Borovicka, T.; Jirina, M., Jr.; Kordik, P.; Jirina, M. Selecting Representative Data Sets. In *Advances in Data Mining Knowledge Discovery and Applications*; BoD: Stafford, VA, USA, 2012; Volume 12, pp. 43–70. [[CrossRef](#)]
33. Finlayson, S.G.; Subbaswamy, A.; Singh, K.; Bowers, J.; Kupke, A.; Zittrain, J.; Kohane, I.S.; Saria, S. The Clinician and Dataset Shift in Artificial Intelligence. *N. Engl. J. Med.* **2021**, *385*, 283–286. [[CrossRef](#)]
34. Pinto, B.; Correia, M.V.; Paredes, H.; Silva, I. Detection of Intermittent Claudication from Smartphone Inertial Data in Community Walks Using Machine Learning Classifiers. *Sensors* **2023**, *23*, 1581. [[CrossRef](#)] [[PubMed](#)]
35. Crone, S.F.; Kourentzes, N. Naive Support Vector Regression and Multilayer Perceptron Benchmarks for the 2010 Neural Network Grand Competition (NNGC) on Time Series Prediction. In Proceedings of the The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010. [[CrossRef](#)]
36. Zhang, G.P. Neural Networks for Time-Series Forecasting. *Handb. Nat. Comput.* **2012**, *1*, 461–477. [[CrossRef](#)]

37. Zaroug, A.; Garofolini, A.; Lai, D.T.H.; Mudie, K.; Begg, R. Prediction of Gait Trajectories Based on the Long Short Term Memory Neural Networks. *PLoS ONE* **2021**, *16*, e0255597. [[CrossRef](#)] [[PubMed](#)]
38. Chomiak, T.; Hu, B. Time-Series Forecasting through Recurrent Topology. *Commun. Eng.* **2024**, *3*, 9. [[CrossRef](#)]
39. Kaastra, I.; Boyd, M. Designing a Neural Network for Forecasting Financial and Economic Time Series. *Neurocomputing* **1996**, *10*, 215–236. [[CrossRef](#)]
40. Chomiak, T.; Rasiah, N.P.; Molina, L.A.; Hu, B.; Bains, J.S.; Füzesi, T. A Versatile Computational Algorithm for Time-Series Data Analysis and Machine-Learning Models. *npj Park. Dis.* **2021**, *7*, 97. [[CrossRef](#)] [[PubMed](#)]
41. Mallikarjuna, M.; Rao, R.P. Evaluation of Forecasting Methods from Selected Stock Market Returns. *Financ. Innov.* **2019**, *5*, 40. [[CrossRef](#)]
42. Moreira, F.R.D.S.; Verri, F.A.N.; Yoneyama, T. Maximum Visibility: A Novel Approach for Time Series Forecasting Based on Complex Network Theory. *IEEE Access* **2022**, *10*, 8960–8973. [[CrossRef](#)]
43. Di Narzo, A.F.; Aznarte, J.L.; Stigler, M. TsDyn: Nonlinear Time Series Models with Regime Switching, R package version 11.0.4.1; 2022. Available online: <https://cran.r-project.org/web/packages/tsDyn/tsDyn.pdf> (accessed on 16 April 2024).
44. Gers, F.; Eck, D.; Schmidhuber, J. Applying LSTM to Time Series Predictable through Time-Window Approaches. In *Artificial Neural Networks—ICANN 2001*; Dorffner, G., Bischof, H., Hornik, K., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2130, pp. 669–676.
45. Kourentzes, N. Time Series Forecasting with Neural Networks 2022, R Package “nnfor” version 0.9.9. Available online: <https://cran.r-project.org/web/packages/nnfor/nnfor.pdf> (accessed on 16 April 2024).
46. Han, Z.; Zhao, J.; Leung, K.F.; Ma, F.; Wang, W. A Review of Deep Learning Models for Time Series Prediction. *IEEE Sens. J.* **2021**, *21*, 7833–7848. [[CrossRef](#)]
47. Chomiak, T.; Sidhu, A.; Watts, A.; Su, L.; Graham, B.; Wu, J.; Classen, S.; Falter, B.; Hu, B. Development and Validation of Ambuloso: A Wearable Sensor for Bio-Feedback Rehabilitation Training. *Sensors* **2019**, *19*, 686. [[CrossRef](#)]
48. Chomiak, T.; Pereira, F.V.; Meyer, N.; de Bruin, N.; Derwent, L.; Luan, K.; Cihal, A.; Brown, L.A.; Hu, B. A New Quantitative Method for Evaluating Freezing of Gait and Dual-Attention Task Deficits in Parkinson’s Disease. *J. Neural Transm.* **2015**, *122*, 1523–1531. [[CrossRef](#)] [[PubMed](#)]
49. Chomiak, T.; Xian, W.; Pei, Z.; Hu, B. A Novel Single-Sensor-Based Method for the Detection of Gait-Cycle Breakdown and Freezing of Gait in Parkinson’s Disease. *J. Neural Transm.* **2019**, *126*, 1029–1036. [[CrossRef](#)] [[PubMed](#)]
50. Tahmazian, I.; Watts, A.; Chen, O.; Ferrara, H.J.; McCrimmon, A.; Hu, B.; Chomiak, T. A Wearable Device-Enabled Therapeutic Approach to Improve Joint Attention in Autism Spectrum Disorder: A Prospective Pilot Study. *J. Neural Transm.* **2023**, *130*, 1601–1607. [[CrossRef](#)] [[PubMed](#)]
51. Stolze, H.; Kuhtz-Buschbeck, J.P.; Drucke, H.; Johnk, K.; Illert, M.; Deuschl, G. Comparative Analysis of the Gait Disorder of Normal Pressure Hydrocephalus and Parkinson’s Disease. *J. Neurol. Neurosurg. Psychiatry* **2001**, *70*, 289–297. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.