Treatment outcome prediction using multi-task learning: application to botulinum toxin in gait rehabilitation

Adil Khan, Antoine Hazart, Omar Galarraga, Sonia Garcia-Salicetti, Vincent Vigneron

MDPI

Article

Treatment Outcome Prediction using Multi-task Learning: Application to Botulinum Toxin in Gait rehabilitation

Adil Khan [1,4,†]*, Antoine Hazart [1], Omar Galarraga [2], Sonia Garcia-Salicetti [3] and Vincent Vigneron [1]*

[1] IBISC EA 4526, univ Evry, université Paris-Saclay, France; adil.khan@universite-paris-saclay.fr, vincent.vigneron@univ-evry.fr
[2] UGECAM Ile-de-France, Movement Analysis Laboratory, Coubert, France; omar.galarraga@ugecam.assurance-maladie.fr
[3] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France; sonia.garcia@telecom-sudparis.eu
[4] Department of Computer Science, Sukkur IBA University, Sindh, Pakistan; adilkhan@iba-suk.edu.pk
* Correspondence: adil.khan@universite-paris-saclay.fr
† Current address: IBISC EA 4526, univ Evry, université Paris-Saclay, France

Abstract: We propose a framework for optimizing personalized treatment outcomes for patients with neurological diseases. A typical consequence of such diseases is gait disorders, partially explained by command and muscle tone problems, associated with spasticity. Intramuscular injection of botulinum toxin type A is a common treatment for spasticity. According to the patient's profile, it is important to offer the optimal treatment combination with highest possible benefit-risk ratio. For prediction of Knee and Ankle kinematics after botulinum toxin type A (BTX-A) treatment, we propose: (1) a regression strategy based on a multi-task architecture composed of LSTM models; (2) to introduce medical treatment data (MTD) for context modeling; (3) a gating mechanism to model treatment interaction more efficiently. Proposed models are compared with and without metadata describing treatments, and with serial models. Multi-task Learning (MTL) achieved the lowest Root Mean Square Error (RMSE) (5.60°) for traumatic brain injury (TBI) patients on Knee trajectories and the lowest RMSE (3.77°) for cerebral palsy (CP) patients on Ankle trajectories.Overall, the best RMSE ranges from 5.24° to 6.24° for MTL models. To the best of our knowledge, this is the first time that MTL is used for post-treatment gait trajectory prediction. MTL models outperform serial models, particularly when introducing treatment metadata. The gating mechanism is efficient to model treatments interaction and in improving the prediction of trajectories.

Keywords: Multi-task Learning; Clinical Gait Analysis; Pathological Gait; Deep Learning; Long Short-Term Memory; Botulinum Toxin

Citation: Khan, A.; Hazart, A.; Galarraga, O.: Garcia-Salicetti,S. and Vigneron, V. Title. *Journal Not Specified* **2022**, *1*, 0. https://doi.org/

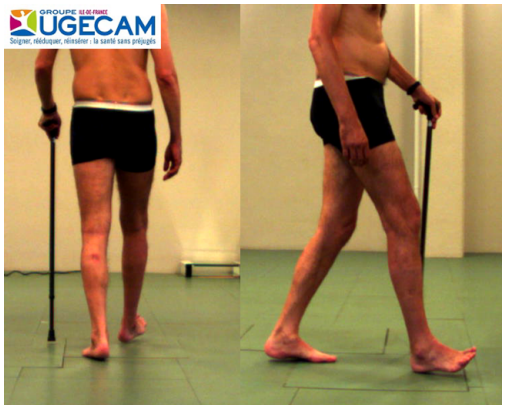
Received:
Accepted:
Published:

1. Introduction

Fatigue, weakness, sensory loss, ataxia, and spasticity are among the usual causes of motor impairments due to neurological diseases such as multiple sclerosis (MS) [1], TBI, spinal cord injury (SCI) or CP, among others. For this reason, people with such impairments are often advised by their physicians to be treated in rehabilitation as a supplement to their background pharmacologic treatment. Spasticity is a motor disorder characterized by a velocity-dependent increase in tonic stretch reflexes (muscle tone) with exaggerated tendon jerks resulting from hyper-excitability of the stretch reflexes as one component of the upper motor neuron syndrome [2]. Intramuscular injection of BTX-A is a standard treatment for spasticity. It has been shown that BTX-A produces improvements in lower and upper limb function [3], thereby improving movement such as walking [4] (see Figure 1) or fine motor skills. The minimum and maximum dose of BTX-A may vary depending on the muscle that is considered [5]. Furthermore, the total dose of BTX-A (sum of doses for all treated muscles) should not exceed a recommended amount according to the patient and the considered muscles (i.e., upper limbs and lower limbs). BTX-A is a relatively expensive

pharmaceutical product and its consumption has increased in recent years [6,7]. Although its effect on muscle function is considered reversible, BTX-A treatment presents risks (i.e., undesirable effect) and injection sessions should be spaced by at least 3 months. For all these reasons, optimizing BTX-A treatment by choosing the right muscles to be treated and the dose distribution is a complex task of great relevance, and requires careful study of the patient's condition.

(a)



(b)

Figure 1. Example of the outcome of BTX-A treatment on gait (a) before treatment (b) after BTX-A treatment.

In practice, decision-making is based on a patient's medical history, physical examination, and Clinical Movement Analysis (CMA). CMA consists in studying movement troubles and identifying their plausible causes, based on biomechanical interpretation of instrumental measures [8] (Figure 2). If certain quality criteria are fulfilled, CMA data are sufficiently reliable for clinical interpretation [9]. CMA techniques can be used to analyze lower limb movement (e.g., walking, climbing stairs, running, etc.) or fine motor skills. Numerous scientific studies have shown that CMA, especially Clinical Gait Analysis (CGA), provides considerable aid in the assessment and treatment decision for various neurological diseases such as CP [10], post-stroke hemiparesis [11], MS [12], among others.

Artificial Intelligence (AI) and Machine Learning (ML) techniques have become almost ubiquitous in our daily lives by supporting or guiding our decisions and providing recommendations. Therefore, it is not surprising that ML approaches are currently becoming more and more popular in precision medicine and fulfill an increasing demand for new healthcare solutions, in particular a better understanding of pathological processes. Among AI and ML methods, deep neural network (DNN) [13] have already shown spectacular results in clinical decision-making aid [14]. DNN require a significant amount of data to be properly trained. However, available experimental databases are often limited in

Figure 2. CMA of lower limbs. Different types of sensors are used to conduct kinematic and kinetic analyses of locomotion in gait labs. These include accelerometers, gyroscopic sensors, magnetometers, force platforms, MoCap systems.

size, which makes these impractical to construct DNNs for prediction models. Medical data is often heterogeneous, complex, incomplete, uncertain, multimodal, and multilevel, which drastically decreases the amount of exploitable data and questions the development of prediction models [15]. ML models must be able to manage data of a different nature describing the patient (images, time series, discrete clinical data, etc.) and link them to data from treatment in nominal, categorical (type of treatment) [16] and/or discrete (doses) forms. This requires that the model be taught a regression task between the data after and before BTX-A treatment. Since these treatments are often a combination of several factors (e.g. several drug injections), it is necessary to be able to model their interactions. Therefore, we propose a strategy to create multitask deep artificial neural networks (DNN). Indeed, MTL can cope with sparse data problems and build a more robust model by sharing knowledge among different tasks [17]. MTL has been widely applied in ML and in the biomedical field, to address the diversity of the data [17].

In the CGA-literature, several works exploited Deep Learning (DL) for predicting gait trajectories, most of them on healthy gait. Su et al. [18] predict gait trajectories and the five gait phases (loading response, mid-stance, terminal stance, pre-swing, and swing) with an Long Short-Term Memory (LSTM) to help in the design of exoskeletons. They employed either 10 or 30-time steps as input for predicting the next five or ten steps. 12 people were enrolled in their experiment and data were collected using attached inertial measurement units (IMUs) on their body parts. Zhu et al. [19] used an attention-based Convolutional Neural Network (CNN)-LSTM to forecast joint trajectories of knee and ankle, based on lower and upper limb data, for the next 60 milliseconds. Zaroug et al. [20] constructed an LSTM auto-encoder to forecast linear acceleration and angular velocity trajectories. To make a prediction of five or ten steps into the future, they considered several lengths of input time steps (five to 40 steps) of kinematic data of six male participants. Hernandez et al. [21] proposed a hybrid network combining an LSTM with a CNN(DeepConvLSTM) to estimate kinematic trajectories, reaching an average Mean Absolute Error (MAE) of 3.6°. Jia et al. [22] constructed a DNN for trajectory prediction using LSTM units and a feature fusion layer. This layer uses EMG and joint angles data. Liu et al. [23] built a deep spatio-temporal model composed of LSTM units to forecast two time-steps into the future, using kinematic data of 35 subjects. More recently, Kolaghassi et al. [24] worked on pathological gait trajectories of children with neurological disorders. They used two deep learning models, an LSTM and a CNN, to forecast hip, knee, and ankle trajectories. Note that all these works tackle the prediction of the same gait cycle. The issue we face in this study is much more complex since it is centered on the impact of several treatments (BTX-A) on gait trajectories.

Our contribution consists in proposing a new solution to predict the BTX-A post-treatment gait trajectory of the patient, and possibly the interaction between different treatments. This solution is a MTL architecture which alleviates the drawbacks previously

mentioned: dataset size (number of patients), sample size (number of features), and features diversity. To the best of our knowledge, this is the first time that MTL is used for post-treatment gait trajectory prediction. This architecture is composed of a collection of LSTM-shaped sub-models, arranged in parallel or in series. Each sub-model is used for one treatment, and each treatment corresponds to an injected muscle. These muscles are attached to the left and right knees and ankles. This MTL model will learn to map pre-treatment gait sequences into post-treatment sequences. A gating mechanism is proposed with different architectures, to control the treatments' influence on the final prediction.

Section 2 presents the data collection and its characteristics. Section 2.3 describes more specifically the different deep architectures used. Most prominent results are presented in section 3. The paper ends with a conclusion and a short discussion.

2. Materials and Methods

2.1. Dataset Acquisition

Data were collected in the Movement Analysis Laboratory of Rehabilitation Center of UGECAM Coubert (France) using a Codamotion system consisting of four CX1 cameras at 100 Hz. All the patients in this laboratory are adults with different types of gait issues. This database consists of patients with central neural system disorders, e.g. CP, SCI, TBI and all patients have undergone spasticity treatment with BTX-A injections.

The database is composed of N_{pat} =38 patients. $N_{uni} = 15$ patients (39.47%) are unilaterally affected (the right lower limb is affected in 6 of them and the left lower limb for the other 9), and $N_{bil} = 23$ patients (60.53%) are bilaterally affected, which means that in total $N_{limbs} = 61$ lower limbs have been modified. The data contains CGA of patients before treatment, medical treatment details, and CGA after treatment. The average age of patients at the time of pre-treatment CGA, time of injection, and the time of post-treatment CGA are 46.67 years old (yo), 46.76 yo, and 46.93 yo respectively. The range of age in the dataset is from 21 to 75 yo. There is approximately a 3-month gap between pre-treatment CGA and post-treatment CGA. Details of patients are listed in Table 1. In this work, we considered injections into four muscles: Soleus, Gastrocnemius (Medialis and Lateralis), Semitendinosus, and Rectus Femoris. We also defined a fifth category called "Other Muscles", which groups all the other muscles that were treated (see Table 2).

There are 28 different combinations of BTX-A injections of these four muscles. A treatment binary code vector

$$\boldsymbol{s}^j = (s_1^j, \ldots, s_c^j)^T, s_i^j \in \{0,1\}, i = 1 \ldots c (c = 5 \text{ as shown in Table 1})$$

is attributed to each lower limb i, with $s_i^j = 1$ if muscle i was injected in limb j, 0 otherwise and $\boldsymbol{d}^j = (d_1^j, \ldots, d_5^j)^T, d_i^j \in \{0,1\}$ is a binary vector put for the disease of patient's limb j. There are five diseases: CP, MS, TBI, SCI, and stroke. ^{T} is the transpose operator.

Table 1. Patient database description.

Total Patients	38
Age (Mean ± SD)	46.76 ± 13.43
Males/Females	24/14
Unilaterally/ Bilaterally affected	15/23
Cerebral Palsy	3
Stroke	9
Multiple Sclerosis	12
Traumatic Brain Injury	3
Spinal Cord Injury	11

Table 2. Considered Injected muscle and their frequencies in the database.

Muscle Number	Muscle/Category	Injections in patient	
		Number	Proportion
1	Soleus	49	29.7%
2	Gastrocnemius Left	37	28.5 %
3	Rectus Femoris	18	10.8%
4	Semitendinosus	12	7.2%
5	Other Muscle	40	24.2 %

2.2. Data Preparation

Kinematic data were automatically segmented into gait cycles from initial contact (IC) to terminal swing (TS), utilizing the high pass algorithm (HPA) [25]. Then gait cycles were resampled and normalized to 51 points (2% of the gait cycle) as proposed by CGA [26], so that DL models are trained with fixed-length sequences as illustrated in Fig. 3. Mean gait cycles were computed for each limb.Combining both pre and post-treatment cycles of each patient leads to a total of $n = 1,622$ gait strides. For any patient's limb j, the input vector is an angular time series $\boldsymbol{x}^j = (x_1^j, \ldots, x_m^j)^T \in [-180, +180]^m$ and the target vector is $\boldsymbol{t}^j = (t_1^j, \ldots, t_m^j)^T$, with $m = 51 \times 2 = 102$. Let $\mathcal{D} = \{\boldsymbol{x}^j, \boldsymbol{t}^j, \boldsymbol{d}^j, \boldsymbol{s}^j\}_{j=1}^n$ be the input-target training set.

The patient's data consists of multiple gait cycles at the time of pre-treatment CGA and post-treatment CGA. Different trials were recorded for each patient. In one trial, there are multiple cycles of pre-treatment CGA. We extract all the cycles of all patients and store them. We separate the right and the left cycles of a person since we consider them as different samples in the data. We perform the same procedure for post-treatment CGA data. Each pre-treatment cycle is associated with a target post-treatment cycle. Note that the number of cycles per patient varies from one patient to another.

There is a total of 5 joints (pelvis, hip, knee, ankle, and foot) and 3 signals per joint in our dataset, leading to 15 signals. These three signals represent the projections of the trajectory of each joint, respectively, on the sagittal, frontal, and transverse planes. In this study, we only consider knee and ankle sagittal planesbecause most treatments are done around these joints.Figures 3a and 3d show the sagittal plane signal (flexion/extension) of ankle and knee for a patient's complete trial containing multiple cycles.

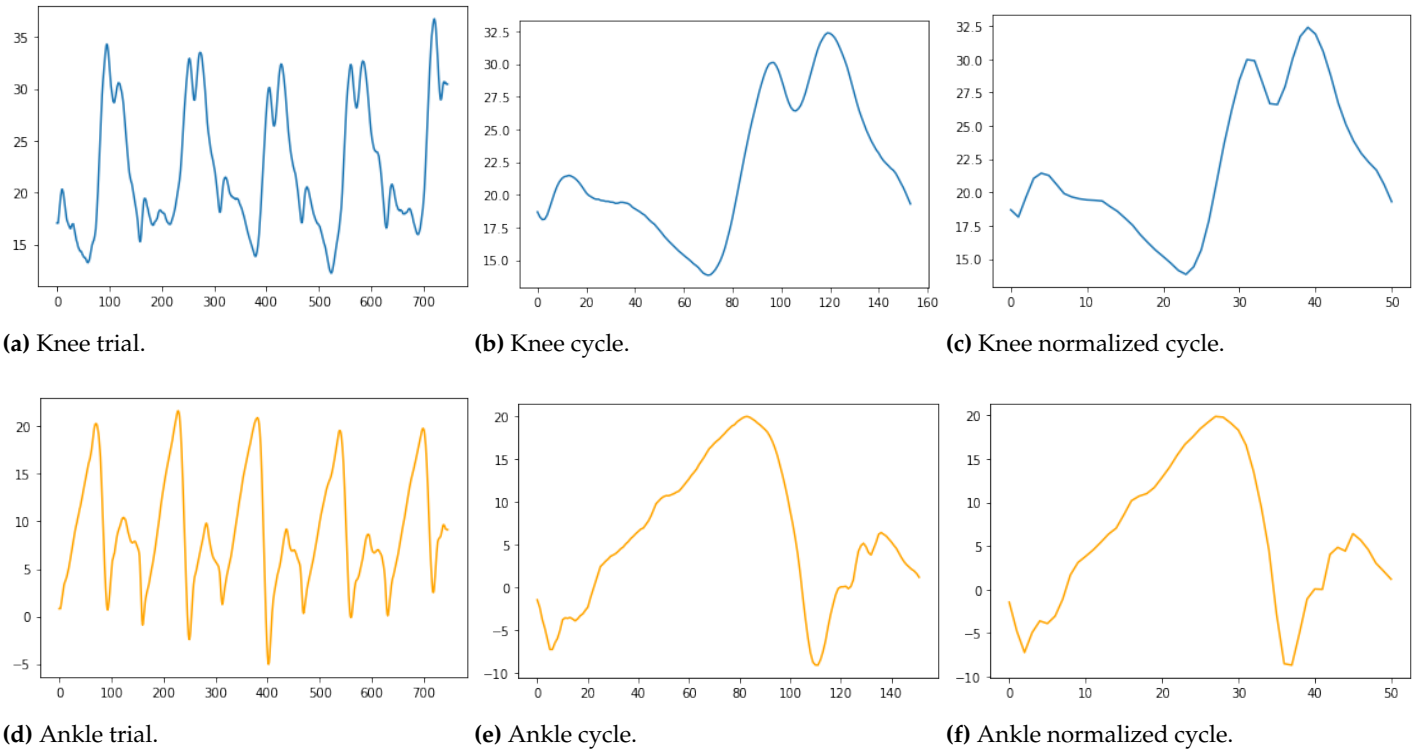
Figures 3b and 3e show a cycle extracted from the full knee and ankle trials, respectively.Figures 3c and 3f show the normalized cycle into 51 points. In the end, our dataset contains 1,622 samples and 210 features: the first feature represents the ID (patient name), the second to 103rd are features of the pre-treatment CGA, then $c = 5$ features describe the presence or absence of botulinum toxin injection according to muscles categories, and finally the last 102 features concern the post-treatment CGA of a patient.

An input matrix X and a target output matrix Y are constructed using the parameters of n training samples, f features (the sagittal plane of the ankle and knee), l_{in} input size, and l_{out} output size. Pre- and post-treatment data were centered and reduced by the standard deviation. The goal is to construct a model with g() that maps $\hat{Y} = g(X)$, where $\hat{Y}$ is a value that is very close to the actual value Y.

2.3. Description of the models

2.3.1. Long Short-Term Memory

When training, early recurrent networks had difficulty remembering information for long periods of time, such as several thousand time steps. Hochreiter et al. [27] introduced a special memory cell capable of retaining information for long periods of time. The LSTM can read and write to its memory. More importantly, this memory never goes through an activation function. This effectively combats the [28] trailing gradient problem and makes the formation of this pattern very stable.



(a) Knee trial. **(b)** Knee cycle. **(c)** Knee normalized cycle.

(d) Ankle trial. **(e)** Ankle cycle. **(f)** Ankle normalized cycle.

Figure 3. Process of converting one trial to one normalized cycle

The original LSTM works with a series of input signals $\boldsymbol{x}_t$. It has a so-called hidden state $\boldsymbol{h}_t$ and cell state $\boldsymbol{c}_t$ of the same size as $\boldsymbol{x}_t$. The cell state $\boldsymbol{c}_t$ is the model's memory. The hidden state $\boldsymbol{h}_t$ is the model's prediction of $\boldsymbol{x}_t$.

The LSTM equations are defined by the following set of matrix equations;

$$\begin{aligned}
\boldsymbol{A} &= \boldsymbol{h}_t \|^1 \boldsymbol{x}_t && (1)\\
\boldsymbol{f}_t &= \sigma(W_f \boldsymbol{A} + \boldsymbol{b}_f) && (2)\\
\boldsymbol{i}_t &= \sigma(W_i \boldsymbol{A} + \boldsymbol{b}_i) && (3)\\
\boldsymbol{o}_t &= \sigma(W_O \boldsymbol{A} + \boldsymbol{b}_O) && (4)\\
\boldsymbol{d}_t &= \tanh(W_d \boldsymbol{A} + \boldsymbol{b}_d) && (5)\\
\boldsymbol{c}_{t+1} &= \boldsymbol{f}_t \circ \boldsymbol{c}_t + \boldsymbol{i}_t \circ \boldsymbol{d}_t && (6)\\
\boldsymbol{h}_{t+1} &= \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_{t+1}) && (7)
\end{aligned}$$

where $\|^1$ is the concatenation operator, $\circ$ is put for the Hadamar product, σ is the logistic function, W are weight matrices and $\boldsymbol{b}$ biases. The basic idea is that the model takes the input $\boldsymbol{x}_t$ and the previous prediction of the current input $\boldsymbol{h}_t$, updates its internal memory $\boldsymbol{c}_t$ to $\boldsymbol{c}_{t+1}$ and then makes a new prediction $\boldsymbol{h}_{t+1}$ based on $\boldsymbol{c}_{t+1}$, $\boldsymbol{h}_t$ and $\boldsymbol{x}_t$.

The original LSTM could have multiple parallel memory cells $\boldsymbol{c}_t$, but as in practice mostly only one memory cell is used; the description of the LSTM was limited to one $\boldsymbol{c}_t$. All the gate functions (equations 2 to 4) are fully connected layers such as $\boldsymbol{y} = f(W\boldsymbol{x} + \boldsymbol{b})$ with a sigmoidal activation function. The data flow in the LSTM is illustrated in Figure 4)

Als,o the role of $\boldsymbol{h}_t$ is not strictly fixed to be a prediction of $\boldsymbol{x}_t$. In fact it can be any series of predictions that is connected to the input series $\boldsymbol{x}_t$. For example, if $\boldsymbol{x}_t$ was the number of people who entered (or left) a building in the last hour, then $\boldsymbol{h}_t$ could be the current number of people inside the building (with an appropriate scaling, so it fits the output range [-1, 1]).

For this study, we have used several variants of LSTM.

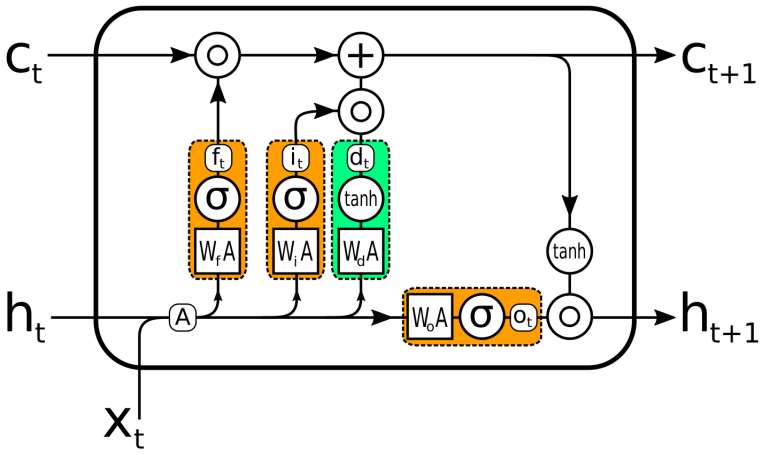


Figure 4. LSTM Unit. The gates which decide which part of the information to pass on are orange. Green is the update to the memory cell.

Five categories of used treatments are reported in Table 2: BTX-A injection of the first four muscles and the fifth category of injections in all other muscles. Each treatment is represented by a LSTM layer. Hidden states represent, according to the DL architecture used, the presence or absence of treatments by BTX-A in the five muscles.

While the LSTM is well suited to prediction tasks on time series, sometimes the knowledge about future events is necessary for a correct prediction. So, the term future is relative to t and means the following data points. Of course, the next/future data points must already be known to be included in the prediction. [29] identified two strategies to integrate knowledge of future events into an LSTM model: bi-directional recurrent neural network (RNN) [30] and delayed input, this second approach consisting in delaying the signal by a delay τ.

Model 1 LSTM is used with pre-treatment CGA data and post-treatment CGA data. Treatments were not considered in this experiment. The model was implemented using five layers of LSTM units, with 51 units per layer, one unit for each point of a cycle. Note that each unit receives a pair of inputs for the knee and ankle respectively. The final layer is fed into a dense layer of 102 neurons (2×51 values), which is then reshaped to get the desired output, shown in Figure 5a. In this model, we initialize the values of the cell state and hidden state of each layer to 0.

Model 2 Total of 5 treatments, together with pre-and post-treatment CGA data, were included in this model, displayed in Figure 5b.In this architecture, the values are initialized according to the medical treatment. If one patient was injected into muscle 1 and muscle 3 (Table 2), then all components of the hidden states vector in LSTM layer 1 and LSTM layer 3 are initialized to 1, and other layers' hidden states vector is initialized as 0. In this model, we also initialize the cell state as 0.

2.3.2. Bi-directional LSTM

The entire signal must be known for this approach. Two LSTM models are trained in parallel, one on the input series (forward) and the other on the reverse input series (backward), starting with the last input and then the forward- last and so on. Thus for each t there are two hidden states $\boldsymbol{h}_{1,t}$ and $\boldsymbol{h}_{2,t}$ among the two available models. $\boldsymbol{h}_{1,t}$ only contains information about the past and $\boldsymbol{h}_{2,t}$ only contains information about the future. Together they have the information about the whole signal and the final prediction $f(\boldsymbol{h}_{1,t}, \boldsymbol{h}_{2,t})$ is made using the two hidden states. This method has the disadvantage that two models have to be trained and therefore the number of parameters and the training time are doubled.

We study the Bi-directional LSTM (Bi-LSTM) architecture and consider two experiments, namely with and without MTD, as previously presented on LSTM.

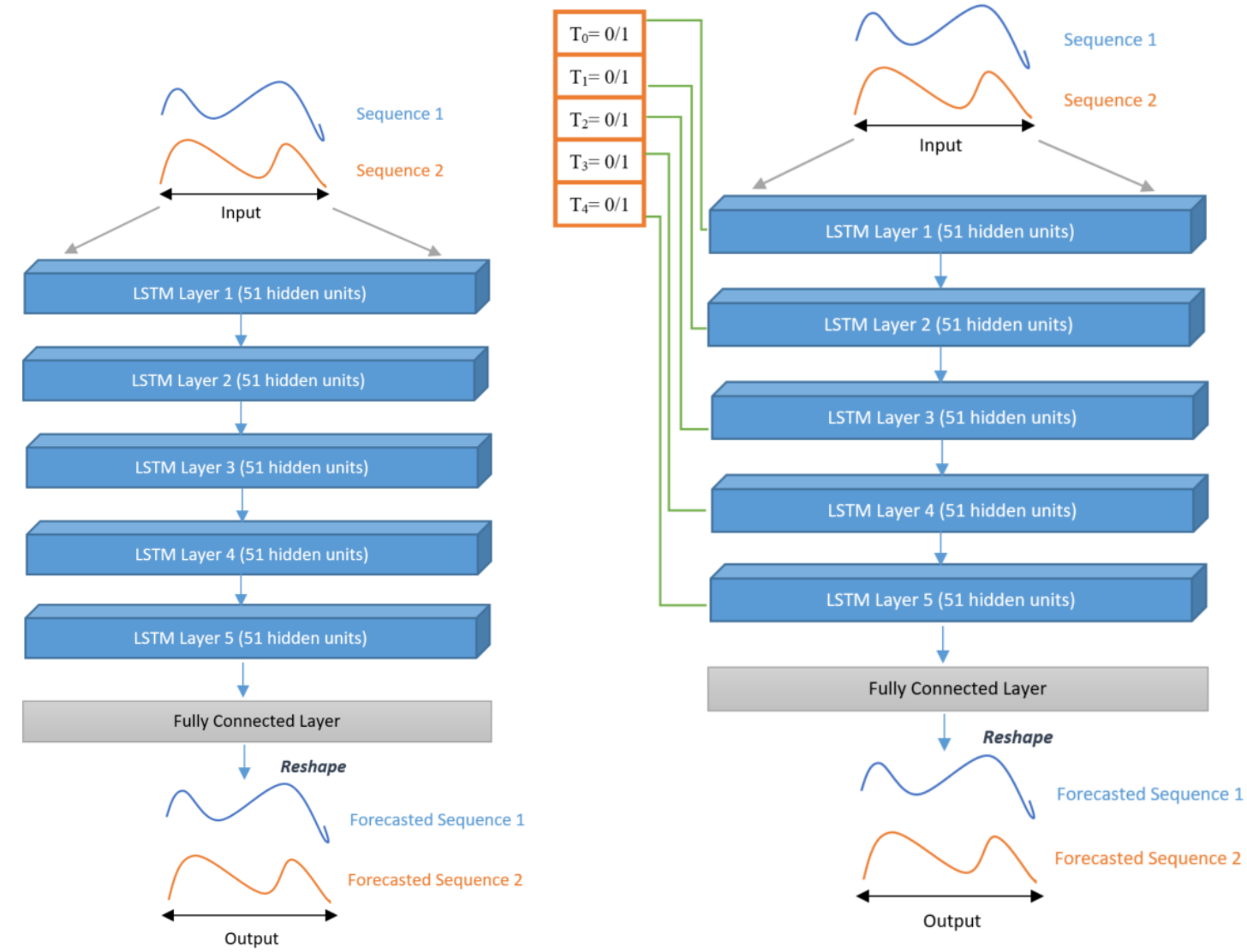


Figure 5. LSTM architectures (Model 1 and Model 2) proposed in this work: (a) without MTD; (b) with MTD

Model 3 is a Bi-LSTM, as depicted in Figure 6. As shown in Fig. 6, the model has mainly the same structure as the previous Model 1 (same number of layers and units in each layer). The final layer's hidden state is fed into a fully connected layer. As in Model 1, we initialize the values of the cell state and hidden state of each layer to 0.

Model 4 This model takes into account MTD in a Multi-task architecture of Bi-LSTM models. Indeed, five Bi-LSTM models work in parallel, while incorporating MTD as in Model 2. Each Bi-LSTM has 51 units, each receiving as input a pair for knee and ankle respectively. Input X is fed to the five Bi-LSTM sub-models, and the cell state of all such sub-models was initialized to 0. Also, the hidden states of all sub-models were initialized according to the presence or absence of MTD (as discussed in Model 2). This architecture has two fully connected layers: the first layer concatenates the outputs of all the sub-models and the second maps the output of the first layer to 102 neurons as per desired output, as shown in Figure 7a.

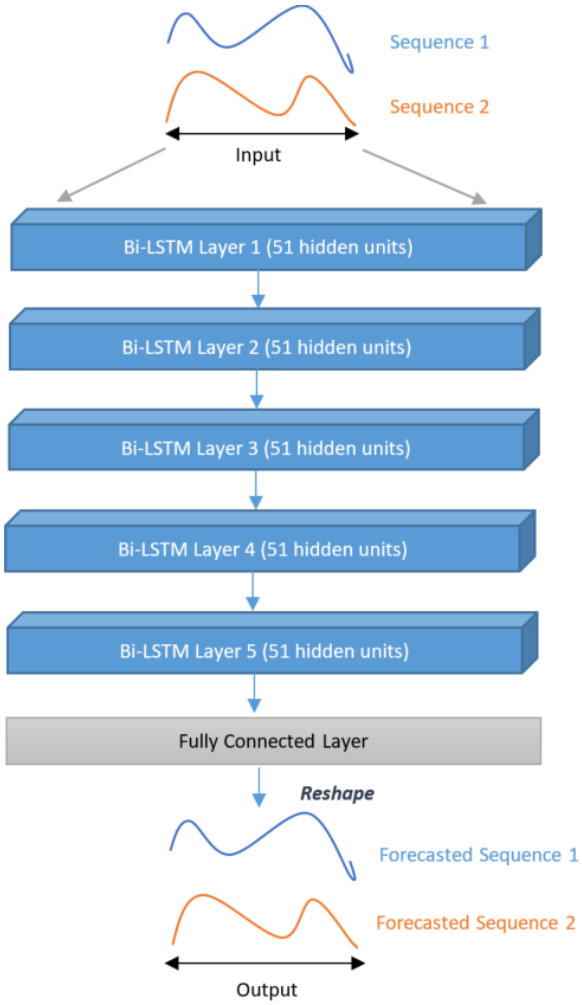


Figure 6. First Bi-LSTM architecture (Model 3) proposed in this work without considering MTD.

Model 5 This model is also a Multi-task architecture of Bi-LSTM sub-models, as in Model 4. But in this case, MTD is considered differently, with a gating mechanism: instead of passing MTD as a hidden state of each Bi-LSTM sub-model, we incorporate them at the end of such sub-models, by multiplying each sub-model's output by its corresponding binary value of MTD. In other words, if there is any treatment, it will be used further in the model; otherwise, it will be discarded (multiplying with 0), as illustrated in Figure 7b. By doing this experiment, we want to assess the impact of this gating mechanism comparatively to MTD internal processing by each sub-model as done in Model 4.

Model 6 & 7 In both models, we replace the first fully connected layer (FC Layer 01) (see Figure 7a and Figure 7b) by a convolutional layer (see Figure 7c and Figure **??**, with kernel size (5,2) and stride (3,2). As there are five Bi-LSTM sub-models and each has an output of size 2×102, we concatenate such outputs and reshape them into a matrix of size (10x102), then given as input to the convolutional layer. Finally, the convolutional layer's output is fed into a fully connected layer of size 102. Model 6 incorporates MTD as in Model 4, through the internal states of sub-models. Model 7 uses the gating mechanism as in Model 5.

2.3.3. Experimental Setup

CGA data consists of 1622 combination pre-treatment and post-treatment gait cycles of 38 patients. Leave-One-Out Cross Validation was used to assess models' performance.

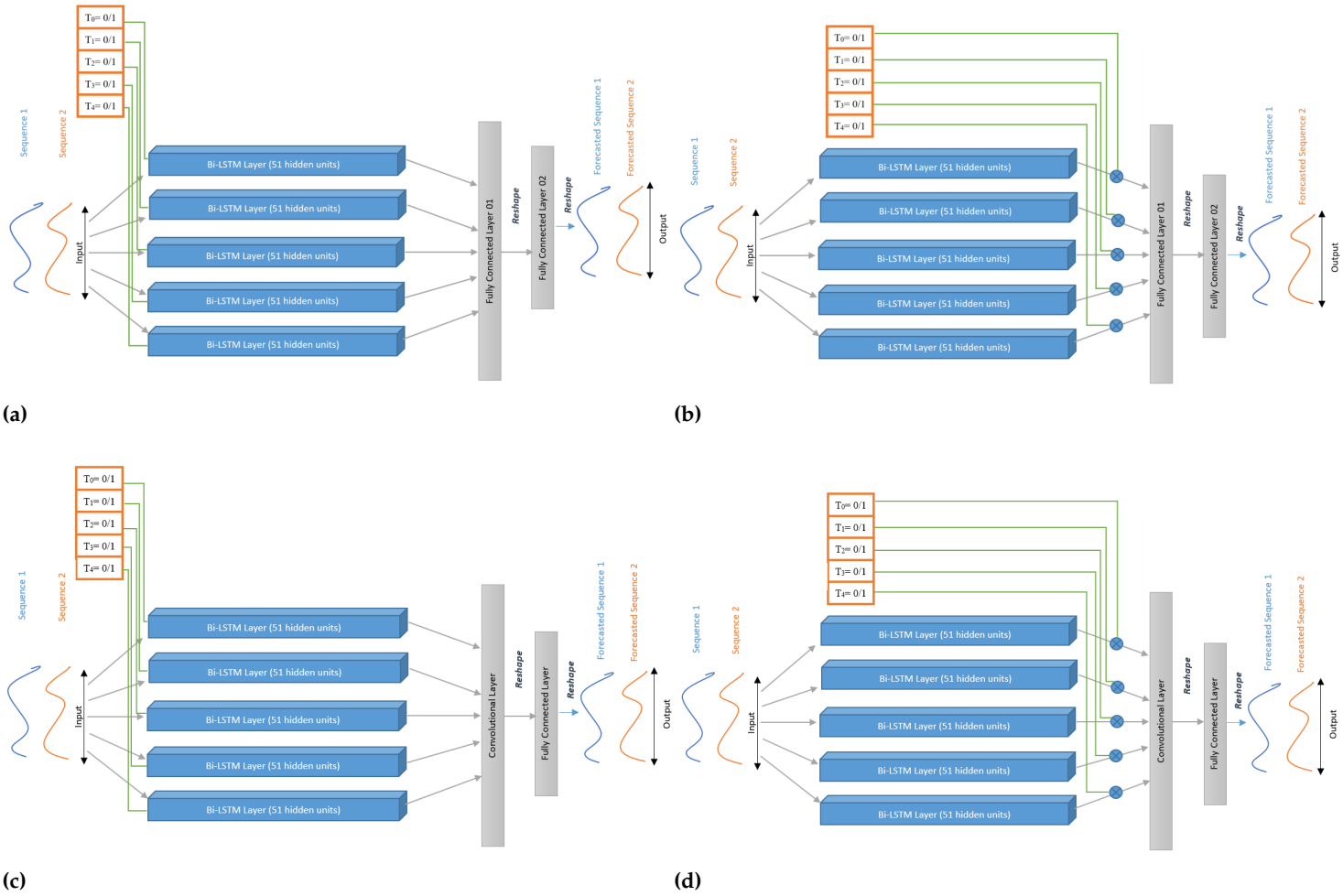


Figure 7. Multi-task Learning architectures with Bi-LSTM sub-models; (a) Model 4: processing MTD internally in each sub-model; (b) Model 5: incorporating MTD through a gating mechanism; (c) Model 6: processing MTD internally in each sub-model using Conv layer; (d) Model 7: incorporating MTD through a gating mechanism using Conv layer

For each iteration, we used 37 patients for training the model and one for testing. In the end, we have taken the RMSE of all tested patients for each model. Mini-batches were used throughout the training process for all models, and the size of each batch was 16. We chose the RMSE as the loss function for optimizing the deep learning models and used the ADAM optimizer for learning. We tried different learning rates and selected the best possible values were selected. We report in Table 3 all details concerning models hyper-parameters.

We calculated the RMSE to see how closely the predicted trajectories of knee and ankle, $\hat{Y}$, match the actual trajectories of knee and ankle, Y. The following equation of RMSE can be derived if we assume that n represents the number of testing samples, f represents the number of features, and l_{out} represents the output size.

$$RMSE = \sqrt{\frac{1}{nfl_{out}} \sum_{i=1}^{n} \sum_{j=1}^{f} \sum_{k=1}^{l_{out}} \left(y_{i,j,k} - \hat{y}_{i,j,k} \right)^2} \tag{8}$$

We also calculated standard error (SE) to measure the variation of RMSE with respect to each disease. SE is calculated using the following formula, where σ represents the standard deviation (SD) of prediction with respect to a particular disease and n represents the total number of patients having a particular disease.

Table 3. Hyper-parameters selection for LSTM, Bi-LSTM, and other variants of architectures. MTD column is used for medical treatment data (included/ not included).

Model No. and Fig. Reference	Model Type	MTD	LSTM layers (units)	Conv. Layer	FC layers (units)	Learning rate
Model 1 (Fig. 5a)	LSTM (Serial)	No	5 layers (51)	None	1(102)	0.005
Model 2 (Fig. 5b)	LSTM (Serial)	Yes	5 layers (51)	None	1(102)	0.005
Model 3 (Fig. 6)	Bi-LSTM (Serial)	No	5 layers (51)	None	1(102)	0.005
Model 4 (Fig. 7a)	MTL, 5 Bi-LSTMs	Yes	1 layer per sub-model (51)	None	2 (1020 & 102)	0.005
Model 5 (Fig. 7b)	MTL, 5 Gated Bi-LSTMs	Yes	1 layer per sub-model (51)	None	2 (1020 & 102)	0.005
Model 6 (Fig. 7c)	MTL, 5 Bi-LSTMs + Conv Layer	Yes	1 layer per sub-model (51)	Kernel(5,2), Stride(3,2)	1(102)	0.005
Model 7 (Fig. 7d)	MTL, 5 Gated Bi-LSTM + Conv Layer	Yes	1 layer per sub-model (51)	Kernel(5,2), Stride(3,2)	1(102)	0.001

$$SE = \frac{\sigma}{\sqrt{n}} \tag{9}$$

We compare and evaluate the performance of the models with the use of these measures.

3. Results

We evaluate models 1 to 7 on our dataset with the above-mentioned metrics and display results in Table 4. These results show the angle difference between actual and predicted trajectories with different architectures. Lowest average RMSE values are displayed in bold; they correspond to the best prediction model according to the diseases reported in Table 4.

From Table 4, we notice that Model 4 outperformed other models in the prediction of post-treatment gait trajectories for patients having MS and TBI. Also, Model 6 performed better for SCI patients than all other architectures. Model 7 outperformed other models of patients having Stroke and CP. We notice that in all cases MTL architectures achieve better performance globally, on both knee and ankle signals.

The following two tables (Table 5 and Table 6) report the difference in angles from actual to predicted gait trajectories of knee and ankle, respectively. In Table 5, the best prediction for Knee angle is obtained for TBI patients by Model 5 with a 5.60°. Also, for all diseases, MTL architectures outperform the others. Model 6 gives the best prediction for MS and SCI; we also note that the gap between Model 6 and Model 7 is prominent. On the other hand, for Stroke patients, Model 7 outperforms the others. In Table 6, we notice that the best RMSE for the ankle is 4.38°, predicted by Model 4, which is lower than that obtained on the knee (5.60°). In case of the ankle, Model 7 gives the best results for SCI 4.49° and again Stroke 6.26°.

Table 4. Performance of different models in prediction of post-treatment gait trajectories with respect to different diseases.

Model No. and Fig. Reference	Model Type	Spinal cord injury (SCI)	Multiple sclerosis (MS)	Stroke	Cerebral palsy (CP)	Traumatic brain injury (TBI)
		No. of patients				
		11	**12**	**9**	**3**	**3**
		No. of Cycles				
		474	**530**	**322**	**148**	**148**
		RMSE Mean ± Standard Error				
Model 1 (Fig. 5a)	**LSTM (Serial)**	6.82±0.09	6.89±0.10	8.11±0.19	7.66±0.14	5.87±0.11
Model 2 (Fig. 5b)	**LSTM (Serial)**	6.71±0.08	6.77±0.08	8.03±0.19	7.23±0.11	7.63±0.32
Model 3 (Fig. 6)	**Bi-LSTM (Serial)**	6.9±0.10	6.38±0.10	7.06±0.18	7.2±0.10	7.78±0.22
Model 4 (Fig. 7a)	**MTL, 5 Bi-LSTMs**	6.26±0.08	**5.8±0.11**	6.99±0.019	6.57±0.12	**5.24±0.13**
Model 5 (Fig. 7b)	**MTL, 5 Gated Bi-LSTMs**	6.67±0.08	6.11±0.09	7.73±0.29	6.22±0.14	6.07±0.21
Model 6 (Fig. 7c)	**MTL, 5 Bi-LSTMs + Conv Layer**	**5.75±0.08**	6.08±0.12	7.16±0.24	6.2±0.12	6.58±0.14
Model 7 (Fig. 7d)	**MTL, 5 Gated Bi-LSTMs + Conv Layer**	6.31±0.12	7.59±0.13	**6.24±0.14**	**6.00±0.14**	7.02±0.07

Bold entries denote the lowest average RMSE over all limbs having a given disease.

Table 5. Performance of different models in prediction of post-treatment knee gait with respect to different diseases

Model No. and Fig. Reference	Model Type	Spinal cord injury (SCI)	Multiple sclerosis (MS)	Stroke	Cerebral palsy (CP)	Traumatic brain injury (TBI)
		No. of patients				
		11	**12**	**9**	**3**	**3**
		No. of Cycles				
		474	**530**	**322**	**148**	**148**
		RMSE Mean ± Standard Error				
Model 1	**LSTM (Serial)**	7.73±0.09	8.05±0.11	8.62±0.21	10.16±0.13	6.66±0.09
Model 2	**LSTM (Serial)**	7.58±0.08	8.26±0.09	7.85±0.17	8.56±0.12	8.05±0.27
Model 3	**Bi-LSTM (Serial)**	8.11±0.13	7.41±0.12	7.77±0.21	7.42±0.11	7.89±0.28
Model 4	**MTL, 5 Bi-LSTMs**	7.51±0.08	7.23±0.14	7.14±0.018	**6.75±0.11**	5.81±0.13
Model 5	**MTL, 5 Gated Bi-LSTMs**	7.62±0.10	7.23±0.11	8.02±0.25	7.00±13	**5.60±0.06**
Model 6	**MTL, 5 Bi-LSTMs + Conv Layer**	**6.94±0.09**	**6.78±0.14**	7.19±0.25	8.63±0.18	8.24±0.17
Model 7	**MTL, 5 Gated Bi-LSTMs + Conv Layer**	8.14±0.12	8.52±0.13	**6.21±0.14**	7.82±0.14	5.94±0.07

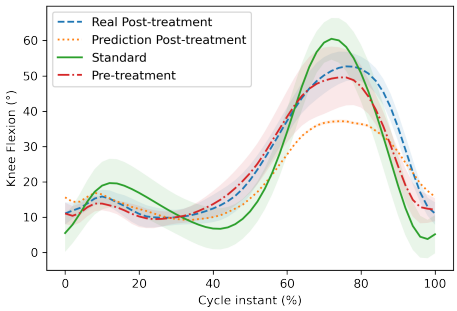
Table 6. Performance of different models in prediction of post-treatment ankle gait with respect to different disease

Model No. and Fig. Reference	Model Type	Spinal cord injury (SCI)	Multiple sclerosis (MS)	Stroke	Cerebral palsy (CP)	Traumatic brain injury (TBI)
		No. of patients				
		11	**12**	**9**	**3**	**3**
		No. of Cycles				
		474	**530**	**322**	**148**	**148**
		RMSE Mean ± Standard Error				
Model 1	**LSTM (Serial)**	5.91±0.08	5.73±0.08	7.61±0.16	5.16±0.14	5.09±0.13
Model 2	**LSTM (Serial)**	5.85±0.07	5.29±0.07	8.21±0.21	5.89±0.10	7.22±0.36
Model 3	**Bi-LSTM (Serial)**	5.69±0.007	5.35±0.07	6.34±0.15	6.99±0.09	5.66±0.15
Model 4	**MTL, 5 Bi-LSTMs**	5.01±0.08	**4.38±0.08**	6.85±0.19	6.4±0.12	**4.68±0.13**
Model 5	**MTL, 5 Gated Bi-LSTMs**	4.56±0.06	5.39±0.10	7.14±0.22	**3.77±0.05**	4.93±0.10
Model 6	**MTL, 5 Bi-LSTMs + Conv Layer**	5.72±0.06	5.00±0.06	7.44±0.32	5.45±0.14	6.54±0.36
Model 7	**MTL, 5 Gated Bi-LSTMs + Conv Layer**	**4.49±0.06**	6.66±0.19	**6.26±0.26**	4.17±0.09	10.63±0.26

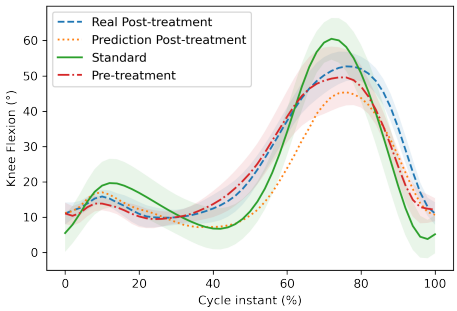
From a different perspective, the following graphs (in Fig. 8 and 9) illustrate the trajectories (pre-treatment, real post-treatment, predicted post-treatment of the patient, and standard trajectory of an adult) of two patients. The Y-axis represents the ankle dorsiflexion or knee flexion, and the X-axis represents the gait cycle of a patient.

Figure 8 compares the prediction of different models on the knee and ankle joints in a patient diagnosed with CP. These figures differentiate the prediction between MTL models and others. Figures 8a, 8b and 8c illustrate the predictions on the Knee angles made by Model 1, Model 2, and Model 3, which are not MTL models. Figure 8d shows the corresponding prediction of Model 7, that is MTL model. Predictions of post-treatment gait from Model 7 are better than others. In other words, it is closer than the expected post-treatment gait trajectory for that patient (average of all his/her target gait cycles in the training set). On the other hand, sub-figures 8e, 8f, 8g, and 8h compare the prediction of the ankle joint of the same patient. Figures 8e and 8f illustrate the prediction of Model 1 and Model 3, respectively. Figures 8g and 8h show the predictions of Model 4 and Model 7, respectively, that are MTL models. We notice that the predicted post-treatment trajectory in the Fig. 8g is better than the first two models that are serial and we see the Fig. 8h improves the prediction significantly at the end of the gait cycle, between 80% and 100%, compared to Fig. 8g. On this patient, MTL models also perform better on the ankle joint.

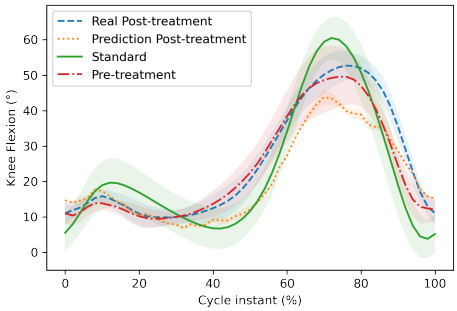
Figure 9 compares the trajectories of the knee and ankle joints of another patient diagnosed with MSs. Figures 9a and 9b, represent the predictions of Knee angles made by Model 1 and Model 2, that are not MTL models. Figure 9c and 9d, represent the prediction of Knee angles made by Model 4 and Model 6 respectively, that are MTL models. We can see clearly that MTL models have better predictions than the first two models. Predicted post-treatment trajectories are closer to real post-treatment trajectories. Last four sub-figures 9e, 9f, 9g, and 9h compare the trajectories of the ankle joint. Figures 9e, 9f, and 9g represent the prediction of Model 1, Model 2, and Model 3. Although Model 3 is not a MTL model, its predictions are much better than the first two serial models. But the prediction of the MTL model (Model 5) in Figure 9h is better than all other models for this particular patient. In general, as proven by Tables (4, 5, and 6), for almost every patient, MTL is performing better.



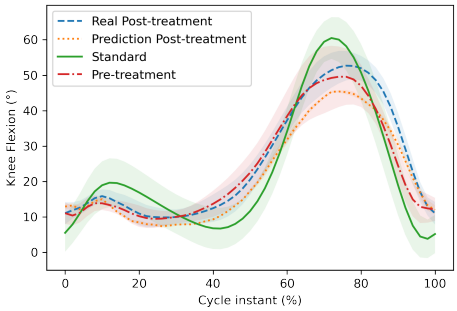
(a) Model 1: LSTM (Serial)



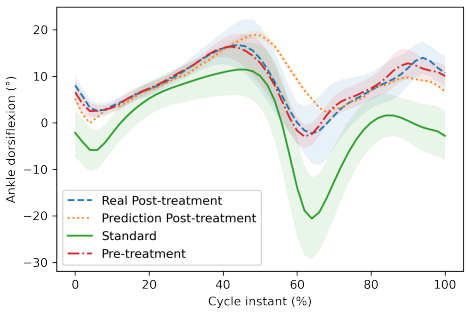
(b) Model 2: LSTM with MTD (Serial)



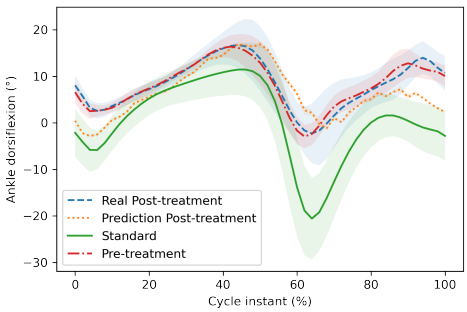
(c) Model 3: Bi-LSTM (Serial)



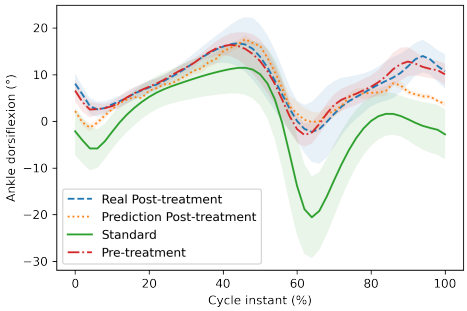
(d) Model 7: MTL, 5 Gated Bi-LSTMs + Conv Layer



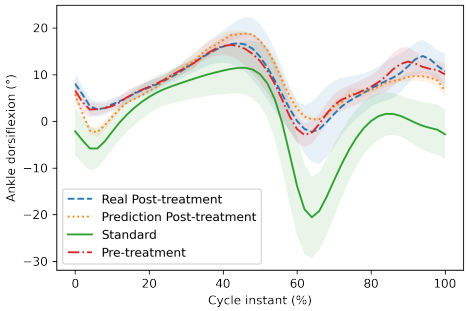
(e) Model 1: LSTM (Serial)



(f) Model 2: Bi-LSTM (Serial)

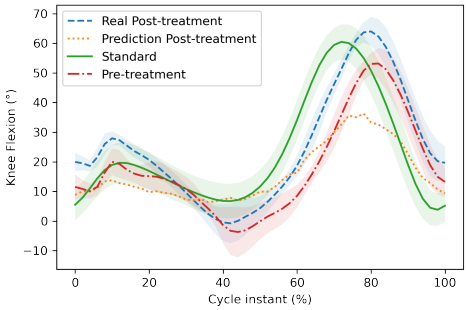


(g) Model 4: MTL, 5 Bi-LSTMs

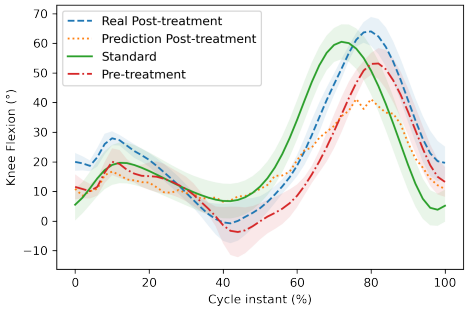


(h) Model 7: MTL, 5 Gated Bi-LSTMs + Conv Layer

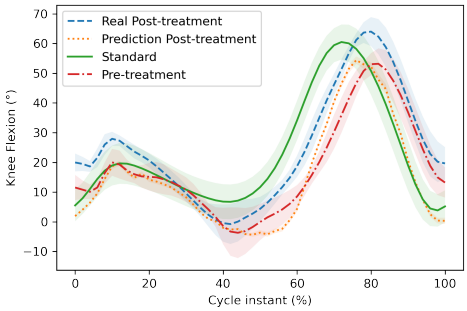
Figure 8. Comparison of the post-treatment gait trajectory of the knee and ankle joint in a patient diagnosed with CP. The first three models (a, b, and c) are serial (Model 1, Model 2, and Model 3), and the fourth model (Model 7) is the MTL model that represents the prediction of the knee joint. Sixth and seventh, two models (e and f) are serial (Model 1 and Model 3), and the last two models (Model 4 and Model 7) are MTL models that represent the prediction of the ankle joint.



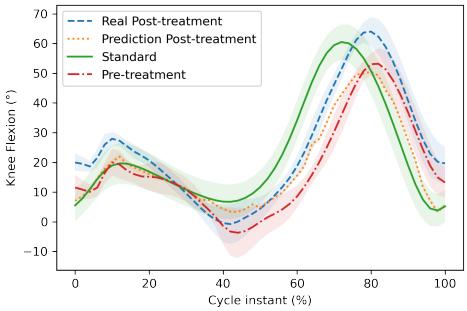
(a) Model 1: LSTM (Serial)



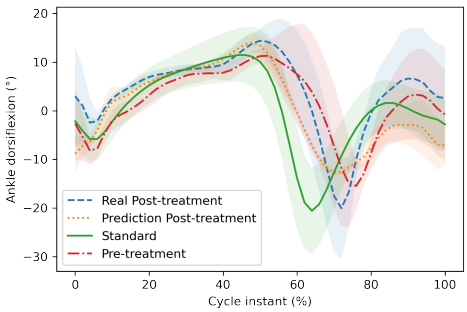
(b) Model 2: LSTM with MTD (Serial)



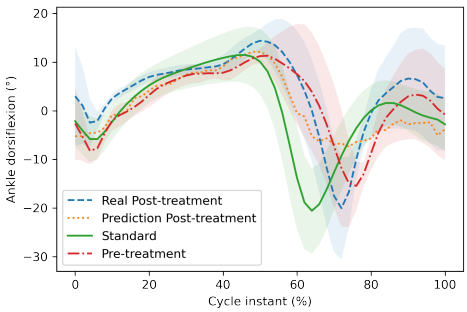
(c) Model 4: MTL, 5 Bi-LSTMs



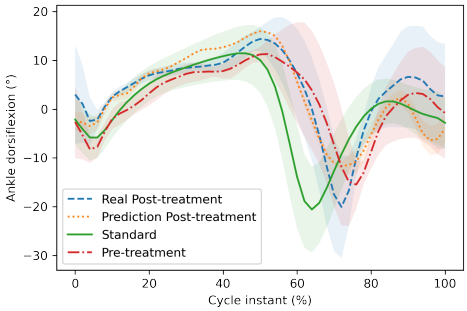
(d) Model 6: MTL, 5 Bi-LSTMs + Conv Layer



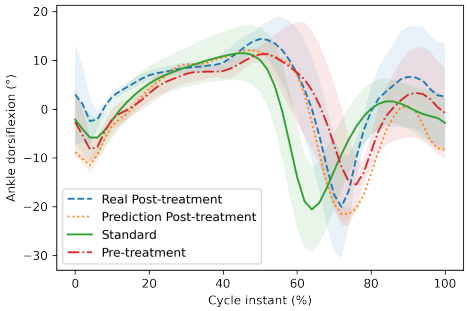
(e) Model 1: LSTM (Serial)



(f) Model 2: LSTM with MTD (Serial)



(g) Model 3: Bi-LSTM (Serial)



(h) Model 5: MTL, 5 Gated Bi-LSTMs

Figure 9. Comparison of the post-treatment gait trajectory of the knee and ankle joint in a patient diagnosed with MS. The first two models (a and b) are serial (Model 1 and Model 2), and the third and fourth models (Model 4 and Model 6) are the MTL model that represents the prediction of the knee joint. Sixth, seventh and eighth, these three models (e, f, and g) are serial (Model 1, Model 2, and Model 3), and the last model (Model 4) is MTL model that represents the prediction of the ankle joint.

4. Discussion and Conclusion

In this study, we use MTL to design an LSTM model and its variants to predict the post-treatment trajectory of adults with abnormal gait. To the best of our knowledge, this specific prediction task, which exhibits greater inter- and intra-subject variability compared to the trajectories of normal adults, has not been addressed before in the literature using MTL.

In order to forecast the trajectories of the knee and the ankle in the sagittal plane, we used LSTM. LSTM was chosen because it has been successfully applied to sequential data and it is able to capture long-term dependencies through its learning [31]. To better evaluate the performance of MTL on a given problem, we also implemented serial models using LSTM as well. RMSE was used to compare the results of both sorts of models. The RMSE of MTL models was lower for all types of patients (different pathologies). We can conclude that MTL models perform better than serial models in our problem consisting of multiple tasks (treatments). MTL architectures allow introducing the medical treatment metadata into the model. Instead of performing a simple post-pre regression task, our results imply that introducing the treatment information (i.e., muscles treated by BTX-A) contributes to better performance.

Overall, the best prediction was obtained for TBI using Bi-LSTM with MTL (Model 4) architecture. Results in Table 4 show that there is only a 5.24° average difference in actual and predicted trajectories. The best maximum average RMSE error between actual and predicted trajectories was 6.24° for stroke patients, using the MTL architecture with gated Bi-LSTM and a convolutional layer (Model 7). For the knee and ankle separately, the best results are 6.75° and 3.77° respectively for CP patients. Even though the proposed method is not tested on the same database, these performances are better than the postoperative predictions in cerebral palsy reported by Galarraga et al. [16], which are 9.0° and 7.5° for knee and ankle gait trajectories respectively, using multiple linear regression.

It is concluded from the results that the number of patients and type of disease do not directly affect the performance of the model. More precisely, we can say that inter- and intra-subject variability affect the performance of the model more than the number of patients (samples) and type of disease. Table 1 gives a detailed description of the number of patients with each disease and Tables 4,5, and 6 report the number of training samples. The minimum number of patients is 3 with CP and TBI diseases, while the maximum number of patients is 12 in MS disease. We notice that the RMSE of CP patients and TBI patients are 6.00° and 5.24°, respectively. On the other hand, the RMSE of MS patients is 5.8°. This shows that having four times more patients for a given disease compared to others doesn't lead to a great difference in the RMSE value.

Finally, Bi-LSTM combined with MTL are highly effective at increasing the total quantity of information that is accessible to the model, hence enhancing the context that is provided to the algorithm. Future work will focus on MTL models with Bi-LSTM networks for exploiting more precise information about treatments, such as the doses information, for further enhancing the context given to the model.

Author Contributions: Adil Khan was the main writer of the article. He also worked in data extraction alongside Omar Galarraga.
Antoine Hazart implemented the models and prepared most of the results.
Omar Galarraga provided the raw data and work on the data preparation. He also gave the clinical and gait analysis insight and prepared some of the images presented in the paper.
Sonia Garcia-Salicetti and Vincent Vigneron provided advice on machine learning and directed the project.
All authors proofread the article

Funding: This work was part of a Master's thesis funded by FéDeV and a Ph.D. project funded by HEC Pakistan

Institutional Review Board Statement:

Informed Consent Statement:

Data Availability Statement:

Acknowledgments: The authors would like to thank the staff of the Movement Analysis Laboratory of UGECAM Coubert, who acquired all the data used in this work

Conflicts of Interest:

Sample Availability:

Abbreviations

The following abbreviations are used in this manuscript:

CMA	Clinical Movement Analysis
CGA	Clinical Gait Analysis
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
MTL	Multi-task Learning
Single-task Learning (STL)	Single-task Learning
DNN	Deep Neural Network
MS	multiple sclerosis
CP	Cerebral Palsy
SCI	Spinal Cord Injury
TBI	Traumatic Brain Injury
BTX-A	Botulinum Toxin type A
CNN	Convolutional Neural Network
RNN	recurrent neural network
LSTM	Long Short-Term Memory
RMSE	Root Mean Square Error
Standard Error (SE)	Standard Error

References

1. McLoughlin, J.; Barr, C.; Crotty, M.; Lord, S.; Sturnieks, D. Association of postural sway with disability status and cerebellar dysfunction in people with multiple sclerosis: a preliminary study. *International journal of MS care* **2015**, *17*, 146–151.
2. Blumhardt, L. *Multiple Sclerosis Dictionary*; Taylor & Francis, 2004.
3. Sun, L.C.; Chen, R.; Fu, C.; Chen, Y.; Wu, Q.; Chen, R.; Lin, X.; Luo, S. Efficacy and Safety of Botulinum Toxin Type A for Limb Spasticity after Stroke: A Meta-Analysis of Randomized Controlled Trials. *BioMed Research International* **2019**, *2019*, 1–17. doi:10.1155/2019/8329306.
4. Roche, N.; Boudarham, J.; Hardy, A.; Bonnyaud, C.; Bensmail, D. Use of gait parameters to predict the effectiveness of botulinum toxin injection in the spastic rectus femoris muscle of stroke patients with stiff knee gait. *European Journal of Physical and Rehabilitation Medicine* **2015**, *51*, 361–370.
5. Notice patient - DYSPORT 500 UNITES SPEYWOOD, poudre pour solution injectable - Base de données publique des médicaments.

6. Tribouillard, H. Elaboration d'une methodologie de validation des indications hors autorisation de mise sur le marche de la toxine botulique de type A au centre hospitalo-universitaire de Lille : Exemple du bavage chez l'adulte. PhD thesis, Faculte de Pharmacie, Universite de Lille, 2018.
7. Battagli, D. Utilisation des toxines botuliques aux Hospices Civils de Lyon : Bilan 2015 des indications. PhD thesis, Faculte de Pharmacie, Universite Claude Bernard - Lyon 1, 2017.
8. Baker, R.W. *Measuring Walking: A Handbook of Clinical Gait Analysis*, 1 ed.; MacKeith Press: London, 2013.
9. McGinley, J.L.; Baker, R.; Wolfe, R.; Morris, M.E. The reliability of three-dimensional kinematic gait measurements: A systematic review. *Gait & Posture* **2009**, *29*, 360–369. doi:10.1016/j.gaitpost.2008.09.003.
10. Moon, D.; Esquenazi, A. Instrumented Gait Analysis: A Tool in the Treatment of Spastic Gait Dysfunction. *JBJS Reviews* **2016**, *4*, 1. doi:10.2106/JBJS.RVW.15.00076.
11. Roche, N.; Boudarham, J.; Hardy, A.; Bonnyaud, C.; Bensmail, B. Use of gait parameters to predict the effectiveness of botulinum toxin injection in the spastic rectus femoris muscle of stroke patients with stiff knee gait. *EUROPEAN JOURNAL OF PHYSICAL AND REHABILITATION MEDICINE* **2015**, *51*, 10.
12. Zörner, B.; Filli, L.; Reuter, K.; Kapitza, S.; Lörincz, L.; Sutter, T.; Weller, D.; Farkas, M.; Easthope, C.S.; Czaplinski, A.; et al. Prolonged-release fampridine in multiple sclerosis: Improved ambulation effected by changes in walking pattern. *Multiple Sclerosis* **2016**, *22*, 1463–1475. doi:10.1177/1352458515622695.
13. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. doi:10.1038/nature14539.
14. Topol, E.J. High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine* **2019**, *25*, 44–56. doi:10.1038/s41591-018-0300-7.
15. Barnes, S.; Saria, S.; Levin, S. An Evolutionary Computation Approach for Optimizing Multilevel Data to Predict Patient Outcomes, 2018. doi:https://doi.org/10.1155/2018/7174803.
16. Galarraga C., O.A.; Vigneron, V.; Dorizzi, B.; Khouri, N.; Desailly, E. Predicting postoperative gait in cerebral palsy. *Gait & Posture* **2017**, *52*, 45–51. Accepted 6 November 2016, doi:10.1016/j.gaitpost.2016.11.012.
17. Zhang, Y.; Yang, Q. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* **2021**.
18. Su, B.; Gutierrez-Farewik, E.M. Gait trajectory and gait phase prediction based on an LSTM network. *Sensors* **2020**, *20*, 7127.
19. Zhu, C.; Liu, Q.; Meng, W.; Ai, Q.; Xie, S.Q. An Attention-Based CNN-LSTM Model with Limb Synergy for Joint Angles Prediction. 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2021, pp. 747–752.
20. Zaroug, A.; Lai, D.T.; Mudie, K.; Begg, R. Lower limb kinematics trajectory prediction using long short-term memory neural networks. *Frontiers in Bioengineering and Biotechnology* **2020**, *8*, 362.
21. Hernandez, V.; Dadkhah, D.; Babakeshizadeh, V.; Kulić, D. Lower body kinematics estimation from wearable sensors for walking and running: A deep learning approach. *Gait & Posture* **2021**, *83*, 185–193.
22. Jia, L.; Ai, Q.; Meng, W.; Liu, Q.; Xie, S.Q. Individualized Gait Trajectory Prediction Based on Fusion LSTM Networks for Robotic Rehabilitation Training. 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2021, pp. 988–993.
23. Liu, D.X.; Wu, X.; Wang, C.; Chen, C. Gait trajectory prediction for lower-limb exoskeleton based on Deep Spatial-Temporal Model (DSTM). 2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM). IEEE, 2017, pp. 564–569.
24. Kolaghassi, R.; Al-Hares, M.K.; Marcelli, G.; Sirlantzis, K. Performance of Deep Learning Models in Forecasting Gait Trajectories of Children with Neurological Disorders. *Sensors* **2022**, *22*, 2969.
25. Desailly, E.; Daniel, Y.; Sardain, P.; Lacouture, P. Foot contact event detection using kinematic data in cerebral palsy children and normal adults gait. *Gait & Posture* **2009**, *29*, 76–80. doi:10.1016/j.gaitpost.2008.06.009.
26. Schwartz, M.H.; Rozumalski, A. The Gait Deviation Index: a new comprehensive index of gait pathology. *Gait & posture* **2008**, *28*, 351–357.
27. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.
28. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

29. Salehinejad, H.; Baarbe, J.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent Advances in Recurrent Neural Networks. *ArXiv* **2018**, *abs/1801.01078*.
30. Schuster, M.; Paliwal, K. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on* **1997**, *45*, 2673 – 2681. doi:10.1109/78.650093.
31. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.