



# Data-driven gait model for bipedal locomotion over continuous changing speeds and inclines

Bharat Singh<sup>1</sup> · Suchit Patel<sup>2</sup> · Ankit Vijayvargiya<sup>3,4</sup> · Rajesh Kumar<sup>3</sup>

Received: 9 April 2022 / Accepted: 2 May 2023 / Published online: 27 May 2023  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Trajectory generation for biped robots is very complex due to the challenge posed by real-world uneven terrain. To address this complexity, this paper proposes a data-driven Gait model that can handle continuously changing conditions. Data-driven approaches are used to incorporate the joint relationships. Therefore, the deep learning methods are employed to develop seven different data-driven models, namely DNN, LSTM, GRU, BiLSTM, BiGRU, LSTM+GRU, and BiLSTM+BiGRU. The dataset used for training the Gait model consists of walking data from 10 able subjects on continuously changing inclines and speeds. The objective function incorporates the standard error from the inter-subject mean trajectory to guide the Gait model to not accurately follow the high variance points in the gait cycle, which helps in providing a smooth and continuous gait cycle. The results show that the proposed Gait models outperform the traditional finite state machine (FSM) and Basis models in terms of mean and maximum error summary statistics. In particular, the LSTM+GRU-based Gait model provides the best performance compared to other data-driven models.

**Keywords** Data-driven model · Kinematic trajectory · Gait model · Loss function · Wire-frame

## Abbreviation

DNN	Deep neural network
LSTM	Long short term memory
GRU	Gated recurrent units
BiLSTM	Bidirectional LSTM
BiGRU	Bidirectional GRU
COMBO BI	BiLSTM + BiGRU
FSM	Finite state machine
MAE	Mean absolute error
MSE	Mean squared error
SE	Standard error

✉ Bharat Singh  
bharatsingh1993@yahoo.com

Suchit Patel  
kingsuchitpatel@gmail.com

Ankit Vijayvargiya  
ankitvijayvargiya29@gmail.com

Rajesh Kumar  
rkumar.ee@mmit.ac.in

- <sup>1</sup> Department of Electrical Engineering, MNIT, Jaipur, India
- <sup>2</sup> Center for Converging Technologies, University of Rajasthan, Jaipur, India
- <sup>3</sup> Department of Electrical Engineering, MNIT, Jaipur, India
- <sup>4</sup> Department of Electrical Engineering, SKIT, Jaipur, India

## 1 Introduction

Several types of mobile robots have been studied and designed in the literature (Singh et al., 2021; Kim et al., 2021; Gao et al., 2021; Burman and Kumar, 2021), which can be categorized by their way of locomotion in different environments and applications: (a) land-based: legged or wheel-type, (b) water-based, and (c) air-based. The main focus of this paper is on land-based mobile robots, and particularly humanoid/bipedal robots that have recently attracted the attention of researchers and industry professionals. This is not only due to their human-like shape, but also their ability to provide efficient and robust walking in uneven terrain, jumping, or running (Wang et al., 2014; Aoi and Tsuchiya, 2011). However, the primary focus of research on bipedal robotics is to achieve human-like walking stability with robustness to disturbances in complex real-world environments (Hosseinmemar et al., 2019; Kwon and Park, 2009; Yamaguchi and Takanishi, 1997; Morris et al., 2019; Doerschuk et al., 2002). The locomotion stability of these robots is hindered by unfavorable conditions such as ground surface inclination or obstacles in the path (Kim et al., 2021; Hildebrandt et al., 2019; Kuffner et al., 2002), making it a great challenge to produce a gait that can ensure stable walking. The

main focus of this work is to deal with the generation of a kinematic trajectory for walking on varied ground slopes at multiple speeds.

In the literature, several successful strategies have been developed for trajectory generation, such as the inverted pendulum (Kim et al., 2020; Razavi et al., 2019; Liu et al., 2021; Sun et al., 2021; Kormushev et al., 2019), the linear inverted pendulum (Huang and Yeh, 2019; Kim et al., 2018; Li et al., 2013; Lee et al., 2016), LSTM pattern generation (Li et al., 2020), multi-objective meta-heuristic optimization (Juang and Yeh, 2017), and fuzzy control (Li et al., 2010). However, these generated trajectories are only applicable to laboratory environments due to issues such as high energy consumption, robustness issues in the presence of disturbances, and difficulty in finding new trajectories in emergency situations. Some authors have applied the learning-based algorithms to generate reference trajectories that have been successfully tested in real-world environments, demonstrating good adaptability to disturbances and irregular terrains (Castillo et al., 2022; Siekmann et al., 2021).

According to the literature, some researchers have used real human locomotion kinematic datasets collected under different conditions as reference trajectories (Holgate et al., 2009; Wang et al., 2021; Liu et al., 2020; Wang et al., 2018; Zhou et al., 2017). This is because human trajectories are optimal and stable in nature. A common strategy from gait analysis, known as a finite state machine (FSM), has been employed to approximate the continuation of multiple tasks and phases (Simon et al., 2014). The strategy is based on dividing the gait cycle into multiple phases and designing separate controllers for each phase. A high-level topology then selects the suitable controller from the set based on the classifier algorithm output, which indicates the phase of the gait cycle. However, as the number of activities increases, so does the requirement for controllers, which requires a lot of expertise that is hard to find. Therefore, authors have used the Discrete Fourier Transform (DFT)-based Gait model to parametrize the continuous kinematic trajectory as a function of the phase variable and tasks (Quintero et al., 2018; Embry et al., 2018). Likewise, authors have developed the Basis model, where the phase variable plays an important role (Embry et al., 2016). The phase variable is a special signal that monotonically increases with the gait cycle, and Tasks is a stacked vector that consists of ground slope and speed. This Gait model helps in predicting the joint kinematic angle with many combinations of speed and incline (Embry et al., 2016). However, the presented model could not capture the relationship between the joints (knee, hip, and ankle), as it has three separate setups for predicting each joint angle individually. Since there is a strong relationship between the joints (Huang et al., 2021), it is necessary to develop a coupled model that can predict the joint trajec-

tory simultaneously while taking the joint relationships into account.

Data-driven models such as deep neural networks, long short-term memory, gated recurrent neural networks, etc., can be employed to find the relationship between the features (Yang et al., 2021; Shrestha and Mahmood, 2019). In our case, the features are the joint knee, ankle and, hip angle. In previous works (Singh et al., 2021, 2022), authors have developed machine learning and deep learning models to capture the important relationship between the joints. However, the developed models were confined only to flat ground with one speed. Additionally, the choice of the objective function for training the data-driven model is not straightforward because it deeply impacts the prediction performance. It has been shown in the literature that mean absolute error outperforms mean squared error for the prediction ability of deep neural networks (Qi et al., 2020). Therefore, a new objective has been developed, taking inspiration from previous studies in the literature (Embry et al., 2016; Qi et al., 2020).

This study proposes using data-driven deep learning approaches to model the kinematics of joint trajectories by using continuously changing inclines and speeds of locomotion data. The dataset consists of locomotion data from 10 able-bodied individuals with continuously changing inclines (-10 to +10 degrees) and speeds (0.8 to 1.2 m/s). A new objective function is also proposed that incorporates the effect of the standard error from the mean trajectory that needs to be followed. This incorporation helps the model avoid high variance points in the dataset. This loss function is used to train the data-driven models. The impact of changing speed with continuously varying inclines with monotonically increasing phase variables is also studied by developing the wireframes. Overall, the major contribution of this study is:

- Data-driven based Gait models are proposed for continuous parametrization of kinematic trajectory as a function of stack tasks vector and monotonically increasing phase variable.
- A new objective function is designed, which incorporates the information of standard error of inter-subject mean trajectory from data, for the training of proposed Gait models.
- Comparative analysis of developed Gait models, based on the mean and maximum error statistics obtained from prediction, for two different cases are briefly discussed.
- Impact of changing inclines with phase variable are studied in detail.
- Wire-frames are also presented from the proposed gait models to find the suitable model for biped robot gait trajectory generation.

This research work is organized as follows: Sect. 2 presents the proposed methodology, which includes the data

description, gait model, and objective function. Then, Sect. 3 explains the results and discussion, which includes the model parameter settings, performance evaluation index, prediction analysis, and the impact of the objective function on the performance of the proposed model. Finally, Sect. 4 concludes the work.

## 2 Proposed methodology

This section discusses the proposed methodology to design the data-driven Gait model for predicting the joint kinematic trajectory angle on continuously varying ground slope and speed. The proposed methodology has three components: (a) data description and preprocessing, (b) development of the data-driven model, and (c) implementation of predicted trajectory wire-frame surfaces to the policy for the control of a biped robot/prosthetic leg. The main focus of this section is on component (b). A schematic flow chart of the proposed methodology is presented in Fig. 1

### 2.1 Data formulation and pre-processing

The kinematic locomotion dataset used in this paper was obtained from the IEEE data port provided by Embry et al. Embry et al. (2018). The dataset was produced by capturing the walking patterns of ten subjects using a ten-camera motion system at 100 Hz. Each subject was asked to walk for twenty-seven combinations of speed and incline (known as tasks and denoted by  $\theta_j$  with  $j = 1, 2, \dots, 27$ ), varying at regular intervals of 0.2 m/s and 2.5 degrees between the range of 0.8 m/s to 1.2 m/s and -10 degrees to +10 degrees, respectively. Each gait cycle consists of 150 time steps for each of the twenty-seven combinations. The obtained dataset was then filtered using a 10th order Weltering filter to produce smooth gait cycle trajectories. Afterward, the dynamic plug-in model based on the Newington-Helen Hayes model was applied to the filtered gait cycle. The resulting output is the kinematic and kinetic dataset [45]. Here, a plug-in gait lower body model is used to calculate the joint angles. The output angles from the model for all joint angles are evaluated from the YXZ Cardan angles. Cardan angles are evaluated by comparing the relative orientations of segments based on the parent and child to the joint. For example, the knee angle is evaluated using the femur and untorsioned tibia segments. The resulting joint angles are relative to each other, such as the ankle angle is relative to the knee angle, the knee angle is relative to the hip angle, and the hip angle is relative to the pelvis angle (it is absolute with respect to the laboratory coordinates). Outliers are removed if they were more than three standard deviations away from the mean trajectory  $\bar{X}_{\phi_i, \theta_j}$  at every instant  $i \in (1, 150)$  in each stride.

### 2.2 Gait model designing

This section discusses the procedure for designing the Gait model and loss function. The Gait model  $q(\phi_i, \theta_j)$  is essentially a parametrization function of a normalized gait cycle  $\phi_i$  for  $i \in (0, 1)$  with task  $\theta_j$ . The key assumption is that the dataset obtained from the previous step is continuous and periodic, which smoothly changes with tasks (Macaluso et al., 2021). The proposed Gait model is developed using data-driven approaches. In this study, seven data-driven models have been used to develop the Gait model for predicting kinematic joint trajectories. These models are: (a) Deep Neural Network (DNN), (b) Long-Short Term Memory (LSTM), (c) Gated Recurrent Units (GRU), (d) Bidirectional LSTM (BiLSTM), (e) Bidirectional GRU (BiGRU), (f) LSTM+GRU, and (g) BiLSTM+BiGRU.

#### 2.2.1 Data-driven models

*Deep Neural Network (DNN)* (Canziani et al., 2016) is a type of neural network with multiple hidden layers between the input and output layers. This type of neural network is inspired by the human brain and requires more data to train itself to provide better accuracy. It establishes a non-linear relationship between inputs and outputs by processing the training dataset. The number of layers helps in deriving better high-level logic from the given inputs. A DNN consists of interconnected neural units that are stacked on top of each other, creating a complex structure. It uses fewer parameters to tune manually, making it easier to generate better logic. Therefore, the network has reusable codes that help in generating better results. The prediction model for the DNN is given by (1),

$$q(\phi_i, \theta_j) = R_n(\dots R_2(W^{(2)} R_1(W^{(1)}[\phi_i, \theta_j] + b_1) + b_2)\dots) \quad (1)$$

where,  $R_n$ ,  $W^n$ , and  $b_n$  is function, weights, and bias for  $n$  layer respectively.

*Long-Short Term Memory (LSTM)* (Greff et al., 2016) is a type of RNN but with three additional gates: input, forget, and output. These gates control the flow of information within the model. The input gate controls the input values that enter the memory cell, while the forget gate filters the important values from the memory cell. The output gate controls the output value from the memory cell. LSTM can memorize the time dependency of data by looking back at past values in the long run, which helps build better predictive logic. It also controls the addition of new information into the memory cells. LSTM has an additional advantage because it helps in dealing with the vanishing and exploding gradient problem that exists in RNNs. The mathematical expression for prediction model

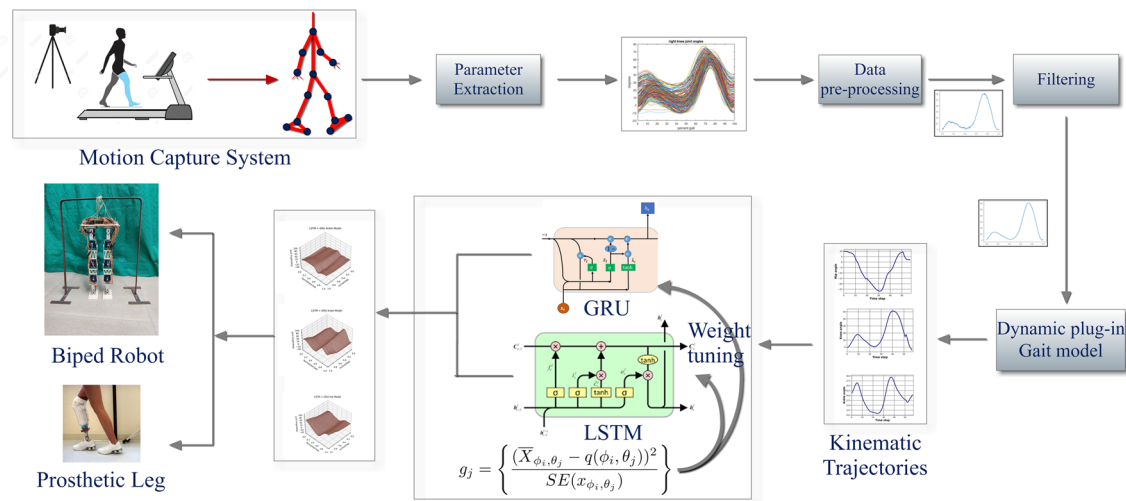


Fig. 1 Schematic flow chart of proposed methodology

using LSTM is presented by (2),

$$q(\phi_i, \theta_j) = R_n \cdot (O_g \cdot \text{relu}(F_g(R_{n-1}) + I_g(\tanh(w_n[h_{t-1}, [\phi_i, \theta_j]] + b_n)))) \quad (2)$$

where,  $R_n$ ,  $W_n$ , and  $b_n$  are function, weight, and bias for  $n$  layer respectively,  $O_g$ ,  $I_g$ , and  $F_g$  are output gate, input gate, and forget gate respectively,  $h_{t-1}$  is input value from previous hidden layer,  $\text{relu}$  and  $\tanh$  are activation function.

*Gated Recurrent Units (GRU)* (Dey et al., 2017) have an LSTM unit with two additional gates: the reset and update gates. The reset gate controls the flow of information, while the update gate controls the update in values of weight and bias. GRU has a hidden state for controlling the flow of information. Essentially, it has an uncontrolled exposure of content of memory cells from the previous step, while LSTM has control over the exposure of content. GRU controls the output values of the layer but doesn't control the addition of new knowledge in the memory cells. The major advantage of GRU is that it takes less training time compared to LSTM. It can easily capture different time dependencies for short times. Therefore, the performance of GRU is better than LSTM if long text and short data in sequence are available, and vice versa. Expression for prediction model is presented by (3),

$$q(\phi_i, \theta_j) = R_n \cdot (\text{relu}(F_g(R_{n-1}) + I_g(\tanh(w_n[h_{t-1}, [\phi_i, \theta_j]] + b_n)))) \quad (3)$$

where,  $R_n$ ,  $W_n$ , and  $b_n$  are function, weight, and bias for  $n$  layer respectively,  $I_g$ , and  $F_g$  are input gate, and forget gate respectively,  $h_{t-1}$  is input value from previous hidden layer,  $\text{relu}$  and  $\tanh$  are activation function.

*Bidirectional LSTM (BiLSTM)* (Huang et al., 2015) is an update to LSTM with an additional bidirectional layer. Due

to its bidirectional nature, the weights and biases are trained using both forward and backward propagation, with the training data flowing alternatively in both directions. This layer captures the relationship between different features available in the dataset. This is an improvement over traditional LSTM, which uses only forward propagation for logic building. The mathematical expression is presented by (4),

$$q(\phi_i, \theta_j) = \left[ \text{LSTM}([\phi_i, \theta_j], \vec{h}_{t-1}), \times \text{LSTM}([\phi_i, \theta_j], \overleftarrow{h}_{t-1}) \right] \quad (4)$$

*Bidirectional GRU (BiGRU)* (Dong, 2018) is a GRU layer with an additional Bidirectional layer. It is a sequence processing model that takes input through forward and backward propagation alternatively. BiGRU is better than normal GRU because it has feed-forward and back-propagation for better logic creation regarding the relationship between the features. It has only input and forget gates similar to GRU. Thus, the forward and backward feeding of inputs helps it to produce better weight and bias values for each input feature. Mathematically, it is given by (5),

$$q(\phi_i, \theta_j) = \left[ \text{GRU}([\phi_i, \theta_j], \vec{h}_{t-1}), \text{GRU}([\phi_i, \theta_j], \overleftarrow{h}_{t-1}) \right] \quad (5)$$

A combination of LSTM and GRU models yields the *LSTM+GRU* model. It captures the advantage of LSTM's ability to capture the relationship of long dataset with short text and GRU's capability of finding the relationship of short dataset with long text. Therefore, it performs better for both types of datasets. It averages the predicted values of LSTM and GRU models. Similarly, a combination of BiLSTM and BiGRU yields the *BiLSTM+BiGRU* model, which lever-



ages the bidirectional nature and LSTM+GRU property to predict the output. It also yields the average of predicted values obtained from respective models. However, the BiLSTM+BiGRU performs better than the LSTM+GRU if the availability of training dataset is more. Both of these models combine the advantage of memory element and conventional method, which is clearly evident in the results and discussion section.

### 2.2.2 Objective function

This section discusses about the comparative analysis between mean squared error and mean absolute error function with their mathematical expression, which is employed in literature for training of models based on multi-variate regression. Afterward, the authors discuss about the rationale behind choosing the new loss function for training of the data-driven model.

*Lipschitz continuity* (Mangasarian and Shiau, 1987; Gouk et al., 2021): A function  $g$  is said to be  $\alpha$ -Lipschitz for all variables  $x, y \in \mathbb{R}^n$ , if it satisfy (6)  $\forall h \geq 1$ ,

$$\|g(x) - g(y)\|_h \leq \alpha \|x - y\|_h \quad (6)$$

*Mean absolute error* (Coyle and Lin, 1988): It quantifies the mean absolute difference between the actual and  $M$  prediction vectors  $A = \{x_1, x_2, \dots, x_M\}$  and  $A^* = \{y_1, y_2, \dots, y_M\}$  respectively, mathematically it is represented by (7),

$$L_{MAE}(A, A^*) = \frac{1}{M} \sum_{j=1}^M \|x_j - y_j\|_1 \quad (7)$$

where,  $\|\cdot\|_1$  is the  $L_1$  norm.

*Mean squared error* (Allen, 1971; Chai and Draxler, 2014): It quantifies the quadratic rule based difference between the actual  $M$  prediction vectors  $A = \{x_1, x_2, \dots, x_M\}$  and  $A^* = \{y_1, y_2, \dots, y_M\}$  respectively, mathematically it is represented by (8),

$$L_{MSE}(A, A^*) = \frac{1}{M} \sum_{j=1}^M \|x_j - y_j\|_2^2 \quad (8)$$

where,  $\|\cdot\|_2$  is the  $L_2$  norm.

Based on the Lipschitz continuity equation (1), it can be easily proved that the MAE and MSE loss function are 1-Lipschitz and not Lipschitz continuous respectively. It is proved in literature, that if function is Lipschitz continuous then it can upper bound the estimated regressor error from the Empirical Rademacher Complexity (Qi et al., 2020). It is suggested that the Laplacian distribution based loss function which relates to MAE can provide the better prediction

then the Gaussian based loss function which relates to MSE. It is true only if the variance related term are same in both expression (Chai et al., 2019).

Mathematical expression of MAE in terms of variance at every-point in stride is (9),

$$L_{MAE}(A, A^*) = \frac{1}{M} \sum_{i=1}^M \left\{ \frac{(\bar{X}_{\phi_i, \theta_j} - q(\phi_i, \theta_j))}{SE(x_{\phi_i, \theta_j})} \right\} \quad (9)$$

Mathematical expression of MSE in terms of variance at every-point in stride is (10),

$$L_{MSE}(A, A^*) = \frac{1}{M} \sum_{i=1}^M \left\{ \frac{(\bar{X}_{\phi_i, \theta_j} - q(\phi_i, \theta_j))^2}{(SE(x_{\phi_i, \theta_j}))^2} \right\} \quad (10)$$

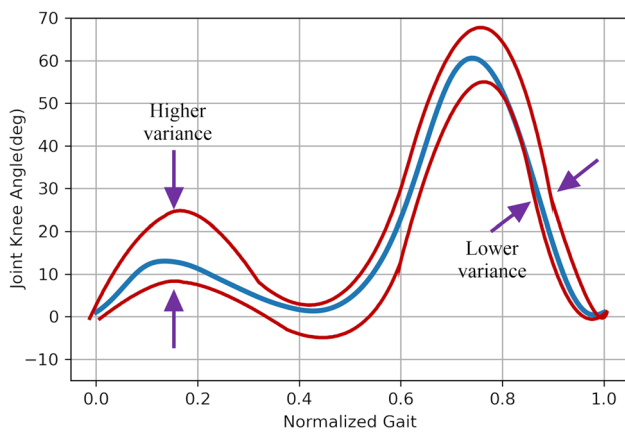
As per literature, MSE loss function can outperform the MAE based loss function if the expected error satisfy the Gaussian distribution with enough samples available for training purpose (Chai and Draxler, 2014). In this paper, authors have less training samples. Thereby, authors have modifies the loss function by integrating the MAE and MSE. It is used for tuning of the weights for data-driven model. The objective function is the square of error between actual and predicted value divided by the standard error in real values at every stride. Since, impact of error is very different in each point of stride, which is incorporated in terms of SE term in expression. Mathematically, it is defined by (11),

$$g_j = \frac{1}{M} \sum_{i=1}^M \left\{ \frac{(\bar{X}_{\phi_i, \theta_j} - q(\phi_i, \theta_j))^2}{SE(x_{\phi_i, \theta_j})} \right\} \quad (11)$$

It provides better precision against the two other loss functions which are validated in Sect. 3. Additionally, it also help in deciding to not follow the values with a high variance error in training dataset, which is our assumption for providing the smooth and continuous trajectory, as shown in Fig. 2.

### 2.2.3 Bayesian optimizer

Bayesian optimization is a powerful technique to find the global optimum solution in a smaller number of steps (Pon and KK, 2021). Basically, it includes the prior belief about the objective function and updates its belief by sampling process to get the posterior belief that is better than the previous belief. The model used for approximating the objective function is known as the surrogate model. In our case, the Gaussian process regressor is used as the surrogate model. The sampling area is determined using the acquisition function, which helps in improvement in current estimates by reducing the variance in observation. The basic steps for Bayesian optimization using the Keras tuner are presented as:



**Fig. 2** Sample trajectory from data-set at speed 1 m/s and incline  $0^\circ$  (blue color trajectory show the mean trajectory  $\bar{X}_{\phi_i, \theta_j}$ ; red color represent the distance of standard error SE from the mean trajectory  $SE(x_{\phi_i, \theta_j})$ )

1. Build the surrogate model that minimizes the objective function, the mean square error is chosen as the objective function that directly relates to the hyper-parameters.
2. Different hyper-parameters are sampled based on variance.
3. Evaluate the true objective using the hyper-parameters sampled from step 2.
4. Update the surrogate model based on the minimum error.
5. Steps 2 to 4 are repeated till the optimum solution is obtained.

### 3 Results and discussion

This section includes the performance evaluation of proposed models using the mean and maximum error statistic indices. In this study, the models are trained and tested using the human locomotion dataset which is derived from the walking data on a different combination of speed and incline. Firstly, the reasoning behind the single multiple output model, parameter settings, and prediction evaluation methods are briefly discussed. Afterward, the prediction analysis of the developed models for two cases is discussed. Lastly, the significant finding has been provided which includes the impact of losses on model performance, dispersion of error in both cases, and wire-frame prediction analysis.

#### 3.1 Model's performance settings

In this study, it is considered that the proposed models contain the three inputs and three outputs configuration. Two rationales behind using this configuration are: (a) the relationship between knee, hip, and ankle joint trajectory is captured efficiently. (b) the input features such as the speed and incline

**Table 1** Training dataset for Case 1 (denoted by  $\checkmark$ )

Incline (deg)	-10	-7.5	-5	-2.5	0	2.5	5	7.5	10
Speed (m/s) 0.8	$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$
1		$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$	
1.2	$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$

**Table 2** Training dataset for Case 2 (denoted by  $\checkmark$ )

Incline (deg)	-10	-7.5	-5	-2.5	0	2.5	5	7.5	10
Speed (m/s) 0.8	$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$	$\checkmark$	$\checkmark$
1		$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$	$\checkmark$
1.2	$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$		$\checkmark$

are constant for one complete gait cycle while the time is increasing at each step during that gait cycle. The model will not be able to predict the output if the output configuration does not have the three output configurations. Since it confuses the model in making the appropriate relationship between three inputs and one output. The model predicts a sloppy gait cycle which is increasing with the values of the time. Whereas, when the model was given three outputs for three input values, the model was able to deduce a good relationship between inputs and outputs. It allows the training algorithm to capture the relationship between the three joints i.e., hip, knee, and ankle effectively. This makes the use of a single model to predict all the gait cycle trajectories at the knee, hip, and ankle.

The dataset consists of a combination of twenty-seven tasks for various values of speed and time. The values of all three inputs speed, hip, and ankle were normalized between 0 and 1. Then, the dataset is split into training and testing sets of two different sizes and additionally, the two cases are also formed. The sizes and combinations of training and testing for both cases are explained in further sections. The input shape for models other than the DNN model is reshaped into a 3-D array. The normalization formula for the speed, incline, and time is given by (12),

$$\phi = \frac{t}{T}, \quad \theta = \frac{i}{I}, \quad S = \frac{v}{V} \quad (12)$$

Here the  $\phi$  is normalised Gait,  $\theta$  is normalised incline,  $S$  is normalised speed,  $t$  is time step,  $T$  is total time,  $i$  is incline in range (-10 to +10),  $I$  is total incline gap,  $v$  is walking speed in range (0.6 to 1.4), and  $V$  is total speed gap.

The layer configuration applied to all the models is (3, 3, [128, 512, 16]). The activation function used for hidden layers is *relu* while *linear* is used for the output layer. The hyperparameters were tuned using the Bayesian Optimizer. The optimizer gave 500,  $1e-2$ , and 10 as optimized values for a number of epochs, learning rate, and batch size respectively.

**Table 3** Maximum error statistic: Case 1

Knee trajectory										
Incline (deg) → Speed (m/s) ↓		− 10	− 7.5	− 5	− 2.5	0	2.5	5	7.5	10
DNN	0.8	0.28667	<b>0.33704</b>	0.27289	0.54786	0.42773	<b>0.30048</b>	0.54630	<b>4.20461</b>	0.82322
	1	<b>0.91073</b>	0.35046	<b>0.36280</b>	0.27154	<b>0.45777</b>	0.23981	<b>0.90786</b>	0.66399	<b>2.37738</b>
	1.2	0.28498	<b>0.43919</b>	0.33390	<b>0.44858</b>	0.34607	<b>0.17275</b>	0.51093	<b>1.15922</b>	0.62716
LSTM	0.8	0.32096	<b>0.62689</b>	0.14529	<b>0.38249</b>	0.19999	<b>0.35391</b>	0.33995	<b>2.52109</b>	0.17618
	1	<b>0.65641</b>	0.16917	<b>0.39198</b>	0.15522	<b>0.37155</b>	0.31920	<b>0.63163</b>	0.33865	<b>1.50686</b>
	1.2	0.25279	<b>0.61773</b>	0.32257	<b>0.53476</b>	0.18924	<b>0.39052</b>	0.24169	<b>0.32059</b>	0.22242
GRU	0.8	0.32601	<b>0.48668</b>	0.23425	<b>0.33103</b>	0.19797	<b>0.57864</b>	0.25573	<b>5.27214</b>	0.21712
	1	<b>0.33171</b>	0.22855	<b>0.69846</b>	0.25992	<b>0.39442</b>	0.25263	<b>0.91243</b>	0.22809	<b>2.16839</b>
	1.2	0.38599	<b>0.82679</b>	0.43449	<b>0.40838</b>	0.27879	<b>0.36394</b>	0.17900	<b>0.64808</b>	0.13182
BiLSTM	0.8	0.43014	<b>0.52763</b>	0.56584	<b>0.56203</b>	0.89330	<b>0.75197</b>	0.71227	<b>7.08934</b>	0.49507
	1	<b>0.80737</b>	0.57125	<b>0.64160</b>	0.71239	<b>0.62104</b>	0.57125	<b>0.41122</b>	0.57636	<b>2.12517</b>
	1.2	0.53530	<b>0.52638</b>	0.77099	<b>0.61575</b>	0.54252	<b>0.69605</b>	0.59820	<b>0.57213</b>	0.44878
BiGRU	0.8	0.18819	<b>0.53429</b>	0.21895	<b>0.65554</b>	0.16817	<b>3.31787</b>	0.29751	<b>8.65588</b>	0.30953
	1	<b>0.68093</b>	0.15561	<b>0.44624</b>	0.18064	<b>0.53650</b>	0.15561	<b>0.38594</b>	0.25399	<b>1.40905</b>
	1.2	0.24459	<b>0.80317</b>	0.27825	<b>0.36109</b>	0.18257	<b>0.54533</b>	0.29247	<b>1.12480</b>	0.31351
LSTM + GRU	0.8	0.17481	<b>0.52979</b>	0.13036	<b>0.34764</b>	0.13989	<b>0.44136</b>	0.26244	<b>3.84163</b>	0.18024
	1	<b>0.39800</b>	0.18751	<b>0.44866</b>	0.19250	<b>0.16126</b>	0.18751	<b>0.61555</b>	0.18558	<b>1.82738</b>
	1.2	0.18661	<b>0.57993</b>	0.25187	<b>0.31548</b>	0.18229	<b>0.33607</b>	0.17203	<b>0.26390</b>	0.11706
BiLSTM+ BiGRU	0.8	0.45221	<b>0.21543</b>	0.56354	<b>0.47874</b>	0.76160	<b>1.86991</b>	0.29906	<b>7.64524</b>	0.35073
	1	<b>0.54831</b>	0.35638	<b>0.30140</b>	0.66817	<b>0.47717</b>	0.35638	<b>0.26012</b>	0.40749	<b>1.42779</b>
	1.2	0.29872	<b>0.28205</b>	0.70282	<b>0.21894</b>	0.30211	<b>0.39921</b>	0.28807	<b>0.48700</b>	0.30605
DNN	0.8	0.10737	<b>0.23655</b>	0.13450	<b>0.20194</b>	0.17049	<b>0.31464</b>	0.23540	<b>0.34558</b>	0.31060
	1	<b>0.23966</b>	0.22178	<b>0.22698</b>	0.13871	<b>0.17647</b>	0.09429	<b>0.34955</b>	0.26715	<b>0.43153</b>
	1.2	0.13718	<b>0.29756</b>	0.17121	<b>0.25007</b>	0.20438	<b>0.28048</b>	0.27350	<b>0.28794</b>	0.28269
LSTM	0.8	0.13662	<b>0.21198</b>	0.11509	<b>0.13326</b>	0.11244	<b>0.25967</b>	0.15238	<b>0.40221</b>	0.17084
	1	<b>0.16837</b>	0.16207	<b>0.12493</b>	0.10864	<b>0.09768</b>	0.09944	<b>0.43749</b>	0.16189	<b>0.32484</b>
	1.2	0.12815	<b>0.25146</b>	0.10118	<b>0.13340</b>	0.10853	<b>0.13761</b>	0.12703	<b>0.31458</b>	0.11112
GRU	0.8	0.16740	<b>0.23174</b>	0.10717	<b>0.28185</b>	0.07577	<b>0.21217</b>	0.12475	<b>1.70619</b>	0.21123
	1	<b>0.33671</b>	0.07138	<b>0.21637</b>	0.14078	<b>0.29187</b>	0.18371	<b>0.48016</b>	0.06907	<b>1.21307</b>
	1.2	0.16101	<b>0.28075</b>	0.16454	<b>0.17367</b>	0.20584	<b>0.47314</b>	0.08776	<b>0.25916</b>	0.07643
BiLSTM	0.8	0.14177	<b>0.18374</b>	0.22285	<b>0.38038</b>	0.29431	<b>0.54814</b>	0.37201	<b>0.50362</b>	0.45127
	1	<b>0.33435</b>	0.42133	<b>0.29023</b>	0.33821	<b>0.35289</b>	0.42133	<b>0.50110</b>	0.44129	<b>0.76526</b>
	1.2	0.20365	<b>0.19760</b>	0.25668	<b>0.36154</b>	0.46307	<b>0.58704</b>	0.40243	<b>0.48415</b>	0.47333
BiGRU	0.8	0.05733	<b>0.48158</b>	0.07836	<b>0.17393</b>	0.06821	<b>0.61877</b>	0.08617	<b>0.48460</b>	0.13473
	1	<b>0.32875</b>	0.14696	<b>0.12915</b>	0.08836	<b>0.15097</b>	0.14696	<b>0.38186</b>	0.07966	<b>0.58957</b>
	1.2	0.05437	<b>0.22743</b>	0.08245	<b>0.15863</b>	0.09357	<b>0.61753</b>	0.08083	<b>0.45403</b>	0.13722
LSTM + GRU	0.8	0.15201	<b>0.20781</b>	0.10332	<b>0.20199</b>	0.07993	<b>0.21914</b>	0.13419	<b>0.99224</b>	0.11810
	1	<b>0.20906</b>	0.09115	<b>0.12091</b>	0.11182	<b>0.17098</b>	0.09115	<b>0.25025</b>	0.09520	<b>0.52882</b>
	1.2	0.14458	<b>0.23004</b>	0.10492	<b>0.10828</b>	0.09760	<b>0.30081</b>	0.07211	<b>0.23279</b>	0.06098
BiLSTM + BiGRU	0.8	0.19495	<b>0.29973</b>	0.20967	<b>0.25944</b>	0.14151	<b>0.33553</b>	0.18120	<b>0.39591</b>	0.15554
	1	<b>0.27783</b>	0.14536	<b>0.14767</b>	0.17378	<b>0.22299</b>	0.14536	<b>0.40616</b>	0.19860	<b>0.64053</b>
	1.2	0.11296	<b>0.20957</b>	0.15521	<b>0.22752</b>	0.13758	<b>0.56227</b>	0.14092	<b>0.33662</b>	0.33808

**Table 3** continued

Knee trajectory										
Incline (deg) → Speed (m/s) ↓		− 10	− 7.5	− 5	− 2.5	0	2.5	5	7.5	10
Ankle trajectory										
DNN	0.8	0.51166	<b>0.42676</b>	0.39027	<b>0.67925</b>	0.35099	<b>0.34049</b>	0.35123	<b>0.50285</b>	0.44793
	1	<b>0.46507</b>	0.39099	<b>0.59086</b>	0.47917	<b>0.43939</b>	0.47770	<b>0.41695</b>	0.44997	<b>0.50073</b>
	1.2	0.47570	<b>0.97107</b>	0.57838	<b>0.51805</b>	0.36496	<b>0.49214</b>	0.47793	<b>0.43882</b>	0.41731
LSTM	0.8	0.18916	<b>0.50869</b>	0.14438	<b>0.30701</b>	0.13351	<b>0.48831</b>	0.20136	<b>0.94745</b>	0.17192
	1	<b>0.32880</b>	0.19033	<b>0.20953</b>	0.19709	<b>0.26441</b>	0.21541	<b>0.50499</b>	0.22799	<b>0.66070</b>
	1.2	0.33110	<b>0.48099</b>	0.30477	<b>0.53928</b>	0.13245	<b>0.48105</b>	0.17136	<b>0.49425</b>	0.18982
GRU	0.8	0.23668	<b>0.42083</b>	0.22459	<b>0.24625</b>	0.20549	<b>0.39442</b>	0.18072	<b>0.72809</b>	0.21123
	1	<b>0.45555</b>	0.15831	<b>0.24845</b>	0.17691	0.49301	0.22496	0.65075	0.21872	<b>1.79997</b>
	1.2	0.25354	<b>0.58124</b>	0.22137	<b>0.36982</b>	0.16199	<b>0.51193</b>	0.13292	<b>0.48626</b>	0.29802
BiLSTM	0.8	0.42381	<b>0.30679</b>	0.40072	<b>0.25522</b>	0.33582	<b>0.61989</b>	0.40122	<b>1.77513</b>	0.58135
	1	<b>0.58336</b>	0.32160	<b>0.16997</b>	0.46806	<b>0.49591</b>	0.32160	<b>0.96325</b>	0.36011	<b>0.83298</b>
	1.2	0.53048	<b>0.54072</b>	0.53119	<b>0.26779</b>	0.27575	<b>0.43667</b>	0.31792	<b>0.50354</b>	0.38762
BiGRU	0.8	0.23229	<b>0.37657</b>	0.25542	<b>0.37782</b>	0.19601	<b>3.43975</b>	0.22538	<b>1.07860</b>	0.26594
	1	<b>0.53570</b>	0.21355	<b>0.24838</b>	0.22211	<b>0.49122</b>	0.21355	<b>0.49532</b>	0.25462	<b>0.62776</b>
	1.2	0.23641	<b>0.69670</b>	0.48646	<b>0.48666</b>	0.22344	<b>0.40871</b>	0.26969	<b>0.60523</b>	0.25985
LSTM + GRU	0.8	0.10354	<b>0.29191</b>	0.14272	<b>0.22604</b>	0.14414	<b>0.30956</b>	0.14317	<b>0.82455</b>	0.12328
	1	<b>0.30105</b>	0.13008	<b>0.20827</b>	0.14215	<b>0.19597</b>	0.13008	<b>0.35855</b>	0.14394	<b>1.11209</b>
	1.2	0.11765	<b>0.41442</b>	0.22948	<b>0.36906</b>	0.10490	<b>0.48133</b>	0.14110	<b>0.37269</b>	0.19937
BiLSTM + BiGRU	0.8	0.62576	<b>0.29711</b>	0.54820	<b>0.18384</b>	0.30405	<b>1.67516</b>	0.42734	<b>1.33759</b>	0.29724
	1	<b>0.30942</b>	0.34239	<b>0.20756</b>	0.47566	<b>0.36990</b>	0.34239	<b>0.58220</b>	0.30770	<b>0.63327</b>
	1.2	0.31763	<b>0.27101</b>	0.50391	<b>0.21341</b>	0.21922	<b>0.33580</b>	0.35393	<b>0.39760</b>	0.27919

The best values of performance indices are given in bold

**Table 4** Maximum error statistic: Case 2

Knee trajectory										
Incline (deg) → Speed (m/s) ↓		− 10	− 7.5	− 5	− 2.5	0	2.5	5	7.5	10
DNN	0.8	0.64557	<b>0.86658</b>	0.73799	0.53174	0.66409	<b>0.65181</b>	0.38288	0.46386	0.67009
	1	<b>0.89258</b>	0.35476	<b>0.57830</b>	0.43627	<b>0.40644</b>	0.35476	<b>0.48380</b>	0.46079	0.34461
	1.2	0.61350	<b>0.39976</b>	0.43881	<b>0.38585</b>	0.33113	<b>0.40993</b>	0.37248	<b>0.70037</b>	0.28467
LSTM	0.8	0.21757	<b>0.21623</b>	0.16425	<b>0.39714</b>	0.23199	<b>0.50364</b>	0.29660	0.28512	0.21488
	1	<b>0.30829</b>	0.14410	<b>0.37973</b>	0.12511	<b>0.47071</b>	0.14410	<b>0.38863</b>	0.20741	0.09861
	1.2	0.17636	<b>0.28066</b>	0.13366	<b>0.22425</b>	0.17456	<b>0.48389</b>	0.13022	<b>0.41086</b>	0.07036
GRU	0.8	0.34195	<b>0.67075</b>	0.57707	<b>0.51415</b>	0.88568	<b>1.53938</b>	0.57584	0.40945	0.76903
	1	<b>0.52841</b>	0.61075	<b>0.59073</b>	0.52660	<b>0.80546</b>	0.61075	<b>1.09187</b>	0.52351	0.55282
	1.2	0.74409	<b>1.14677</b>	1.01650	<b>0.81622</b>	0.68312	<b>0.56634</b>	0.43480	<b>0.39209</b>	0.22025
BiLSTM	0.8	0.15674	<b>0.26700</b>	0.07328	<b>0.55360</b>	0.13403	<b>0.42692</b>	0.10577	0.17035	0.15966
	1	<b>0.42696</b>	0.08157	<b>0.17931</b>	0.14008	<b>0.43516</b>	0.08157	<b>1.92425</b>	0.11989	0.08013
	1.2	0.14947	<b>0.28997</b>	0.13640	<b>0.20106</b>	0.09938	<b>0.44017</b>	0.08175	<b>1.02219</b>	0.07372
BiGRU	0.8	0.20423	<b>0.41485</b>	0.10075	<b>0.42689</b>	0.14520	<b>0.82227</b>	0.31879	0.16540	0.10795
	1	<b>0.51777</b>	0.18694	<b>0.30366</b>	0.10081	<b>0.34242</b>	0.18694	<b>0.65883</b>	0.16072	0.13968
	1.2	0.16719	<b>0.34672</b>	0.13285	<b>0.40904</b>	0.15487	<b>0.33391</b>	0.08642	<b>0.96199</b>	0.11924



**Table 4** continued

Knee trajectory										
Incline (deg) → Speed (m/s) ↓		−10	−7.5	−5	−2.5	0	2.5	5	7.5	10
Ankle trajectory										
LSTM + GRU	0.8	0.13074	<b>0.26113</b>	0.24078	<b>0.37947</b>	0.37505	<b>0.76508</b>	0.22788	0.24740	0.33945
	1	<b>0.36864</b>	0.26116	<b>0.24488</b>	0.29254	<b>0.61173</b>	0.26116	<b>0.71560</b>	0.32067	0.29518
	1.2	0.36626	<b>0.61111</b>	0.51245	<b>0.37901</b>	0.29858	<b>0.27396</b>	0.21539	<b>0.24154</b>	0.11434
BiLSTM + BiGRU	0.8	0.11651	<b>0.31844</b>	0.06121	<b>0.47402</b>	0.04248	<b>0.44848</b>	0.14944	0.12061	0.11828
	1	<b>0.44077</b>	0.07000	<b>0.17701</b>	0.07064	<b>0.38188</b>	0.07000	<b>1.16174</b>	0.10491	0.08637
	1.2	0.09859	<b>0.26059</b>	0.07300	<b>0.27981</b>	0.05471	<b>0.19063</b>	0.04823	<b>0.79826</b>	0.07634
DNN	0.8	0.09791	<b>0.28823</b>	0.14505	<b>0.19260</b>	0.16800	<b>0.19890</b>	0.19249	0.46386	0.29189
	1	<b>0.33806</b>	0.25819	<b>0.27449</b>	0.23819	<b>0.23192</b>	0.25819	<b>0.19922</b>	0.20270	0.24273
	1.2	0.17442	<b>0.21616</b>	0.22240	<b>0.24244</b>	0.24723	<b>0.32394</b>	0.29793	<b>0.23406</b>	0.25493
LSTM	0.8	0.05934	<b>0.16502</b>	0.08458	<b>0.21881</b>	0.08031	<b>0.20868</b>	0.08643	0.28512	0.10581
	1	<b>0.25489</b>	0.11611	<b>0.14150</b>	0.11434	<b>0.15317</b>	0.11611	<b>0.20887</b>	0.08325	0.12499
	1.2	0.06216	<b>0.22008</b>	0.10350	<b>0.14674</b>	0.06650	<b>0.21165</b>	0.07174	<b>0.39184</b>	0.12914
GRU	0.8	0.08790	<b>0.22591</b>	0.07553	<b>0.27071</b>	0.10414	<b>0.60900</b>	0.20104	0.40945	0.29165
	1	<b>0.23990</b>	0.29470	<b>0.18843</b>	0.12439	<b>0.19960</b>	0.29470	<b>0.29359</b>	0.21315	0.26077
	1.2	0.10275	<b>0.18302</b>	0.11079	<b>0.22435</b>	0.11750	<b>0.20205</b>	0.16713	<b>0.23522</b>	0.10290
BiLSTM	0.8	0.05029	<b>0.30286</b>	0.03941	<b>0.13643</b>	0.09346	<b>0.30521</b>	0.17013	0.17035	0.17521
	1	<b>0.26919</b>	0.12007	<b>0.12250</b>	0.05100	<b>0.32775</b>	0.12007	<b>0.71831</b>	0.12168	0.07907
	1.2	0.06306	<b>0.11642</b>	0.05446	<b>0.11101</b>	0.12029	<b>0.44051</b>	0.18877	<b>0.49438</b>	0.06129
BiGRU	0.8	0.08776	<b>0.33794</b>	0.06720	<b>0.21499</b>	0.04576	<b>0.58261</b>	0.12221	0.16540	0.16248
	1	<b>0.36093</b>	0.07835	<b>0.13503</b>	0.05400	<b>0.24518</b>	0.07835	<b>0.33659</b>	0.10085	0.12248
	1.2	0.08736	<b>0.21063</b>	0.06506	<b>0.14745</b>	0.05804	<b>0.28136</b>	0.08086	<b>0.32665</b>	0.10844
LSTM + GRU	0.8	0.05249	<b>0.11730</b>	0.04781	<b>0.22186</b>	0.06790	<b>0.21632</b>	0.11017	0.24740	0.16025
	1	<b>0.21272</b>	0.20076	<b>0.15013</b>	0.06436	<b>0.17639</b>	0.20076	<b>0.24448</b>	0.11120	0.11624
	1.2	0.06684	<b>0.17068</b>	0.08926	<b>0.17982</b>	0.07543	<b>0.09288</b>	0.10441	<b>0.30689</b>	0.07677
BiLSTM + BiGRU	0.8	0.03826	<b>0.28709</b>	0.03955	<b>0.14555</b>	0.04093	<b>0.33380</b>	0.10188	0.12061	0.11800
	1	<b>0.29877</b>	0.07596	<b>0.12070</b>	0.02666	<b>0.22080</b>	0.07596	<b>0.47160</b>	0.07420	0.07480
	1.2	0.05392	<b>0.11717</b>	0.02709	<b>0.11622</b>	0.05676	<b>0.36094</b>	0.10657	<b>0.33602</b>	0.06036
DNN	0.8	0.31037	<b>0.39307</b>	0.35736	<b>0.50562</b>	0.37044	<b>0.31188</b>	0.37001	0.46386	0.29189
	1	<b>0.33737</b>	0.32896	<b>0.32696</b>	0.24697	<b>0.29922</b>	0.32896	<b>0.29922</b>	0.23532	0.26886
	1.2	0.27822	<b>0.27752</b>	0.20353	<b>0.35768</b>	0.26422	<b>0.36755</b>	0.22147	<b>0.36453</b>	0.25644
LSTM	0.8	0.14774	<b>0.31590</b>	0.10980	<b>0.22024</b>	0.08881	<b>0.24417</b>	0.08004	0.28512	0.14868
	1	<b>0.21338</b>	0.13644	<b>0.46414</b>	0.14357	<b>0.51618</b>	0.13644	<b>0.51618</b>	0.17171	0.10513
	1.2	0.16905	<b>0.31499</b>	0.14650	<b>0.30178</b>	0.10248	<b>0.27999</b>	0.08723	<b>0.23562</b>	0.14923
GRU	0.8	0.25570	<b>0.26875</b>	0.27436	<b>0.34443</b>	0.39442	<b>0.97983</b>	0.34327	0.40945	0.42904
	1	<b>0.51124</b>	0.28975	<b>0.26558</b>	0.33096	0.36211	0.28975	0.36211	0.32671	0.31190
	1.2	0.25433	<b>0.31172</b>	0.40979	<b>0.35921</b>	0.29636	<b>0.42913</b>	0.56406	<b>0.53081</b>	0.29785
BiLSTM	0.8	0.05418	<b>0.35155</b>	0.06479	<b>0.29473</b>	0.05522	<b>0.69886</b>	0.11785	0.17035	0.17521
	1	<b>0.23766</b>	0.09211	<b>0.16178</b>	0.05731	<b>0.82262</b>	0.09211	<b>0.82262</b>	0.14493	0.09690
	1.2	0.14197	<b>0.27065</b>	0.08548	<b>0.24411</b>	0.06719	<b>0.40380</b>	0.08572	<b>0.87854</b>	0.10897

**Table 4** continued

Ankle trajectory										
Ankle trajectory										
BiGRU	0.8	0.12340	<b>0.29989</b>	0.08983	<b>0.91198</b>	0.12193	<b>0.76094</b>	0.12390	0.16540	0.17025
	1	<b>0.18801</b>	0.08307	<b>0.21175</b>	0.09261	<b>0.41341</b>	0.08307	<b>0.41341</b>	0.21025	0.11565
	1.2	0.16662	<b>0.34379</b>	0.13293	<b>0.20379</b>	0.13798	<b>0.19618</b>	0.09317	<b>0.57964</b>	0.09459
LSTM + GRU	0.8	0.14336	<b>0.20070</b>	0.14868	<b>0.26849</b>	0.17530	<b>0.56671</b>	0.16462	0.24740	0.21281
	1	<b>0.28043</b>	0.15876	<b>0.30937</b>	0.16397	<b>0.34849</b>	0.15876	<b>0.34849</b>	0.21157	0.15894
	1.2	0.15435	<b>0.25574</b>	0.26984	<b>0.17320</b>	0.18443	<b>0.30314</b>	0.32564	<b>0.33599</b>	0.19402
BiLSTM + BiGRU	<b>0.8</b>	0.05144	<b>0.28018</b>	0.04959	<b>0.55082</b>	0.05711	<b>0.35408</b>	0.07935	0.12061	0.11800
	1	<b>0.16585</b>	0.06946	<b>0.13009</b>	0.04796	<b>0.42953</b>	0.06946	<b>0.42953</b>	0.14829	0.07038
	1.2	0.08515	<b>0.28222</b>	0.08021	<b>0.18688</b>	0.08362	<b>0.21184</b>	0.06544	<b>0.66183</b>	0.06089

The best values of performance indices are given in bold

**Table 5** Summary statistic for Case 1

Model	Train						Test					
	Hip		Knee		Ankle		Hip		Knee		Ankle	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max
DNN	0.19637	0.31060	0.42755	0.82322	0.44030	0.57838	0.27992	0.43153	0.97125	4.20461	0.52172	0.97107
LSTM	0.12824	0.17084	0.24238	0.33995	0.20005	0.33110	0.23057	0.43749	0.71588	2.52109	0.47811	0.94744
GRU	0.13192	0.21123	0.25788	0.43449	0.20753	0.29801	0.47360	1.70619	1.03239	5.27214	0.56820	1.79997
Bi-LSTM	0.33789	0.47333	0.60953	0.89330	0.41119	0.58135	0.42231	0.76525	1.22674	7.08934	0.59625	1.77513
Bi-GRU	0.08903	0.14696	0.23492	0.31351	0.25505	0.48646	0.36898	0.61877	1.49666	8.65588	0.75911	3.43975
<b>LSTM+GRU</b>	<b>0.10736</b>	<b>0.15201</b>	<b>0.18224</b>	<b>0.26244</b>	<b>0.14155</b>	<b>0.22948</b>	<b>0.29024</b>	<b>0.99224</b>	<b>0.77743</b>	<b>3.84163</b>	<b>0.42042</b>	<b>1.11209</b>
COMBO BI	0.19392	0.27398	0.27501	0.41238	0.18266	0.36963	0.33244	0.64053	1.12395	7.64524	0.52414	1.67516

The best values of performance indices are given in bold

In order to quantify the performance of the proposed models, performance indices inspired from Embry et al. (2016) are chosen. It measures how closely the model can predict the untrained tasks. The error between the real value of mean trajectory found using the experiments,  $\phi_i, \theta_j$  and predicted kinematic trajectory  $q(\phi_i, \theta_j)$  for tasks  $\theta_j$  is defined by (13),

$$g_j = \max_i \left\{ \frac{\overline{X}_{\phi_i, \theta_j} - q(\phi_i, \theta_j)}{SE(x_{\phi_i, \theta_j})} \right\} \quad (13)$$

For every tasks maximum is evaluated by taking the largest error encountered in a gait cycle. Then, the mean and maximum of above evaluated maximum error for all tasks is calculated, which is used to quantify the model performance.

## 3.2 Prediction analysis

### 3.2.1 Case 1

The dataset is split into 52:48 ratio for training and testing sets. Overall fourteen and thirteen tasks combination are selected for training and testing purpose. Tables 1, 2 presents

the training dataset for all data-driven model denoted by the  $\checkmark$  and other blank are used for testing purpose.

Table 3 presents the maximum error obtained from all proposed models for knee, hip, and ankle joint trajectories. Every entry shows the maximum error encountered in the predicted gait cycle. Here, the bold value represent the combination used for the testing purpose whereas the normal value for training purpose. Tables 4, 5 present the mean and maximum error statistic for proposed model, which is extracted from the Table 3. LSTM and GRU obtain the approximate 40% less error than the DNN model. Even both models outperform the BiLSTM as well. And, also on testing dataset, LSTM and GRU outperform the BiGRU. However, LSTM+GRU provide better statistic than the LSTM and GRU individually for both in training and testing. It is because, the LSTM+GRU is suitable for all type of dataset as discussed in Sect. 2. Due to the lack of data availability, BiLSTM+BiGRU could not perform better than the LSTM+GRU model.

It is inferred from above discussion, precision of all models is comparable for our dataset. The performance of LSTM+GRU suggests that this is the best option to be used

**Table 6** Summary statistic for Case 2

Model	Train						Test					
	Hip		Knee		Ankle		Hip		Knee		Ankle	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max
DNN	0.21657	0.29793	0.47903	0.73799	0.29477	0.46386	0.24900	0.33800	0.57300	0.89300	0.35200	0.50600
LSTM	0.09427	0.12914	0.17846	0.29660	0.13612	0.28512	0.21102	0.39184	0.36946	0.50364	0.30505	0.51618
GRU	0.16005	0.29470	0.58358	1.01650	0.34708	0.56406	0.26107	0.60900	0.78747	1.53938	0.43327	0.97983
Bi-LSTM	0.09425	0.18877	0.11818	0.17035	0.09995	0.17521	0.30405	0.71831	0.56060	1.92425	0.42541	0.87853
Bi-GRU	0.09123	0.16248	0.15332	0.31879	0.13209	0.21025	0.28903	0.58261	0.50348	0.96198	0.39724	0.91197
<b>LSTM+GRU</b>	<b>0.09449</b>	<b>0.20076</b>	<b>0.27802</b>	<b>0.51245</b>	<b>0.19344</b>	<b>0.32564</b>	<b>0.18995</b>	<b>0.30689</b>	<b>0.44110</b>	<b>0.76508</b>	<b>0.29813</b>	<b>0.56671</b>
COMBO BI	0.07887	0.12499	0.11310	0.28512	0.10076	0.28512	0.25533	0.47160	0.44833	1.16174	0.32109	0.66183
BASIS	0.23700	0.25800	0.30600	0.32500	0.22300	0.24200	0.33300	0.69600	0.48500	1.28600	0.35800	0.67600
FSM	0	0	0	0	0	0	0.568	0.906	1.077	3.961	0.765	1.335

The best values of performance indices are given in bold

for the generation of gait cycle trajectory as compared to other models. It works best in capturing the main relationship between input and output of time series dataset.

### 3.2.2 Case 2

From Table 2, it can infer that two combinations of speed 0.8 m/s and 1 m/s and incline at  $7.5^\circ$  and  $10^\circ$  respectively, provide the highest maximum error which were affecting the precision of all data-driven models. To increase the precision, these two combinations of speed and incline is included into training dataset. Thus, the overall training and testing dataset is in ratio of 60:40. Overall Sixteen and eleven tasks combination are selected for training and testing purpose. Table 2 presents the training dataset for all data-driven model denoted by the  $\checkmark$  and other blank are used for testing purpose.

Table 4 discussed the maximum error obtained by the predicted trajectories from each model for twenty-seven combination of speed and incline. The normal values represent the error obtained for prediction on the training dataset and the bold values present the prediction on testing dataset. Every value in table present the maximum error encountered in the predicted gait cycle from the data-driven model.

Table 4 is summarized in Table 6, which discussed the overall statistic of mean and maximum error obtained from all proposed models for all tasks. It presents that the performance of all models is significantly improved by introducing the worst performer tasks combination of case 1 into case 2. In most cases, the LSTM+GRU outperforms all other models in this case well. All models are also compared with the previous work in literature i.e., basis and FSM model (Embry et al., 2016). Chosen parameter settings are the same for both models as in this study. It shows that the proposed model in this study outperforms the basis and FSM model approximately in the range of 20–50 % for all knee, hip, and ankle

trajectories for both training and testing datasets. It is inferred from the above result that the LSTM + GRU model provides high precision as compared to all other models. Therefore, it is recommended to use the LSTM+GRU for providing the reference trajectories.

### 3.3 Discussion

Figures 3 and 4 presents the sample trajectory prediction of knee, hip, and ankle joint angle for both cases at 1 m/s speed for  $0^\circ$  incline, for all data-driven models. It shows that the LSTM+GRU model provides a smooth trajectory as well as accurately follows the actual gait cycle. Therefore, it is recommended to use the combo of LSTM and GRU for gait generation in real-time for providing references to biped robot.

Table 7 shows that the overall statistic of prediction accuracy for all models. It tells that mean and max of error in case II from all models is less than the case I. It is because the training data-set used for case II includes the worst performer incline and speed data (i.e.  $7.5^\circ$  incline at speed 0.8 m/s and  $10^\circ$  incline at speed 0.8 m/s) of case I.

Table 8 shows the impact of different loss function in term of mean and max error statistic for DNN model only. It shows that the our proposed loss function outperform DNN-MAE by more than approximate 20% in most cases. And also, DNN-proposed model based on the proposed loss outperforms the DNN-MSE by large margins. It validate the theoretical background developed in objective function section.

Comparative analysis of trajectory prediction for all developed models is shown in Figs. 5 and 6. One-hundred-fifty points of the normalized incline and gait are used to make the wire-frame plots. Trajectories are predicted for 1 m/s speed. In basic models (DNN, LSTM, GRU, BiLSTM, and BiGRU)

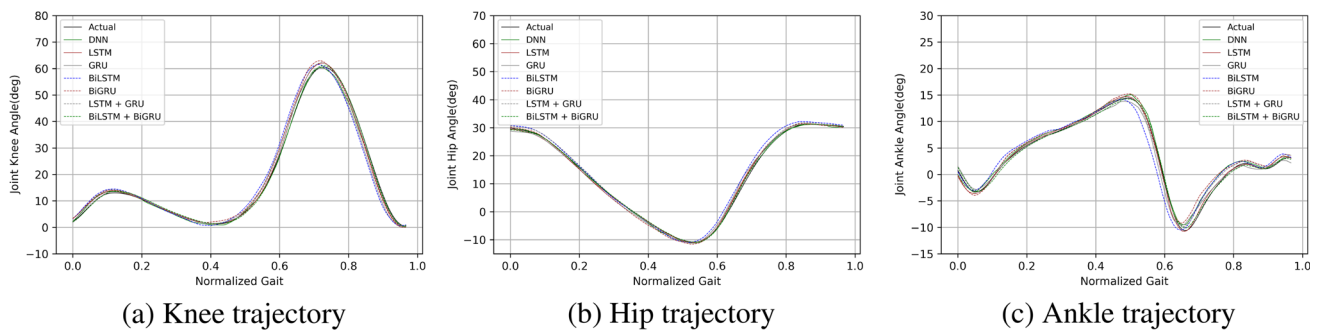


Fig. 3 Comparative analysis of all models for case 1 at speed 1 m/s and 0° incline

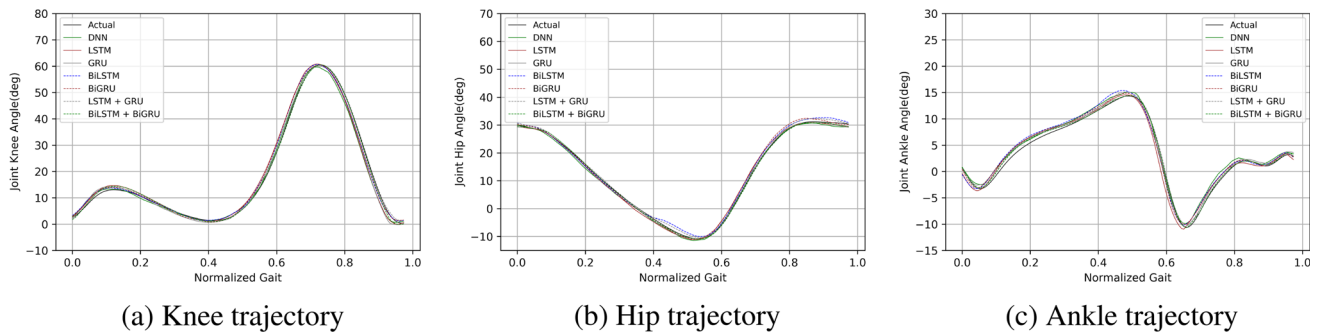


Fig. 4 Comparative analysis of all models for case 2 at speed 1 m/s and 0° incline

Table 7 Summary statistic for case 1 and case 2 (at speed 1 m/s and incline 0°)

Cases	Knee		Hip		Ankle	
	Mean	Max	Mean	Max	Mean	Max
Case 1	0.44483	4.31736	0.24511	2.89211	0.29550	1.24776
<b>Case 2</b>	<b>0.37494</b>	<b>3.49498</b>	<b>0.22416</b>	<b>2.61489</b>	<b>0.27339</b>	<b>1.06537</b>

The best values of performance indices are given in bold

as shown in Fig. 5, the BiLSTM model is performing well because it captures the relationship of speed and incline as compared to all other models. The predicted trajectory from the model smoothly changes from the +10° to -10°. Firstly, the magnitude of the angle decreases from the +10° to 0° and then increases from +0° to -10° incline. As per Fig. 6,

the combination of LSTM+GRU provides smooth trajectory prediction. And, also the combo LSTM+GRU outperforms the BiLSTM model as well in smooth prediction for all joint angles. The performance of the LSTM+GRU model is better than other models because it can memorize the directionality variations in the inclination slope more accurately. The

Table 8 Impact of Loss function on model performance

Model	Train		Knee		Ankle	
	Mean	Max	Mean	Max	Mean	Max
<b>DNN-Proposed</b>	<b>0.196374</b>	<b>0.310601</b>	<b>0.427547</b>	<b>0.823219</b>	<b>0.440299</b>	<b>0.578384</b>
DNN-MAE	0.258133	0.501682	0.577442	0.973513	0.529546	0.923301
DNN-MSE	0.300802	0.490659	0.540519	0.8176	0.586455	0.865027
Test						
<b>DNN-Proposed</b>	<b>0.279918</b>	<b>0.431532</b>	<b>0.971252</b>	<b>4.204612</b>	<b>0.521,724</b>	<b>0.971,071</b>
DNN-MAE	0.561181	0.858816	1.67293	4.000961	0.978883	2.856015
DNN-MSE	1.177479	2.197443	1.594051	3.353269	1.409751	2.410433

The best values of performance indices are given in bold



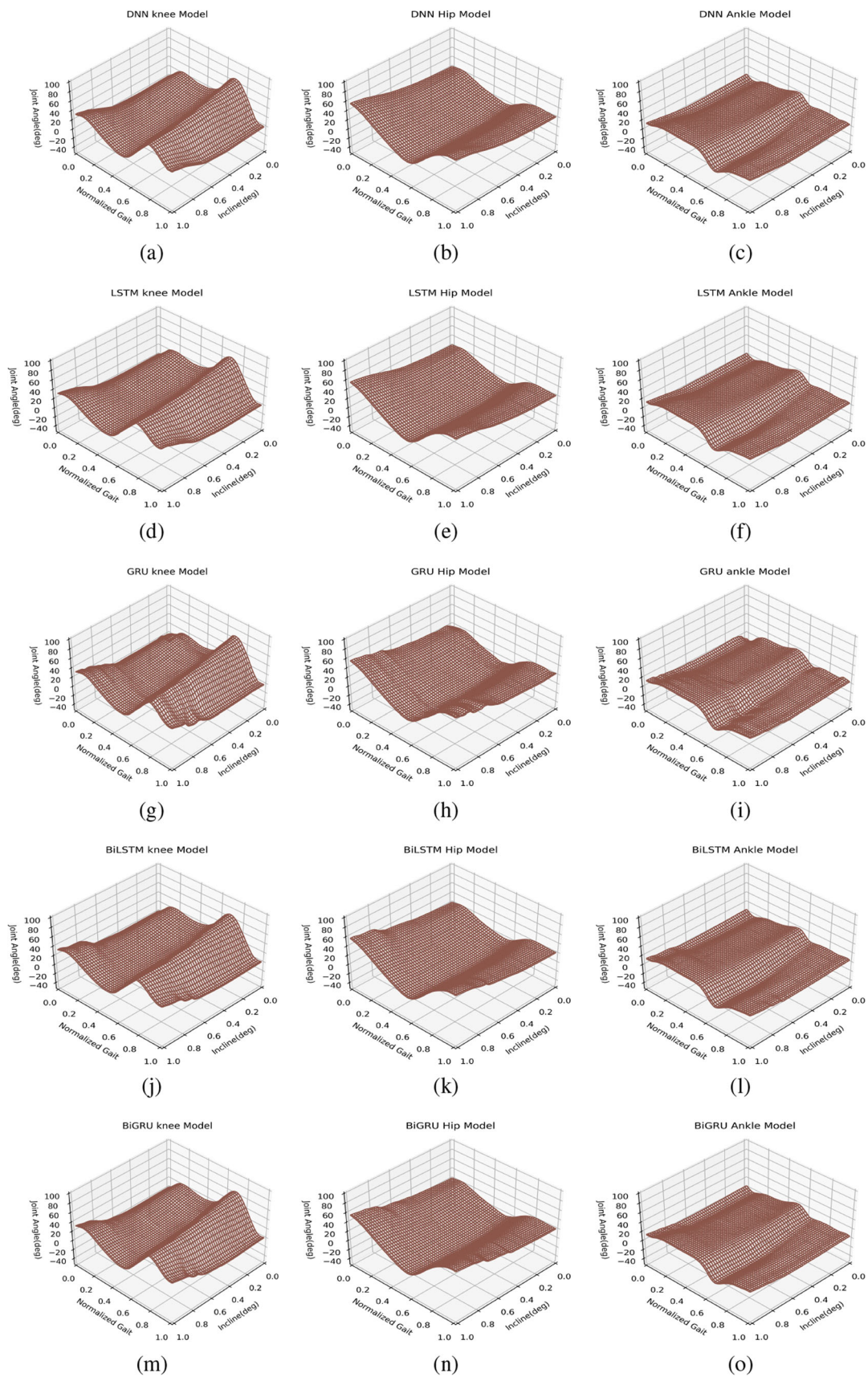
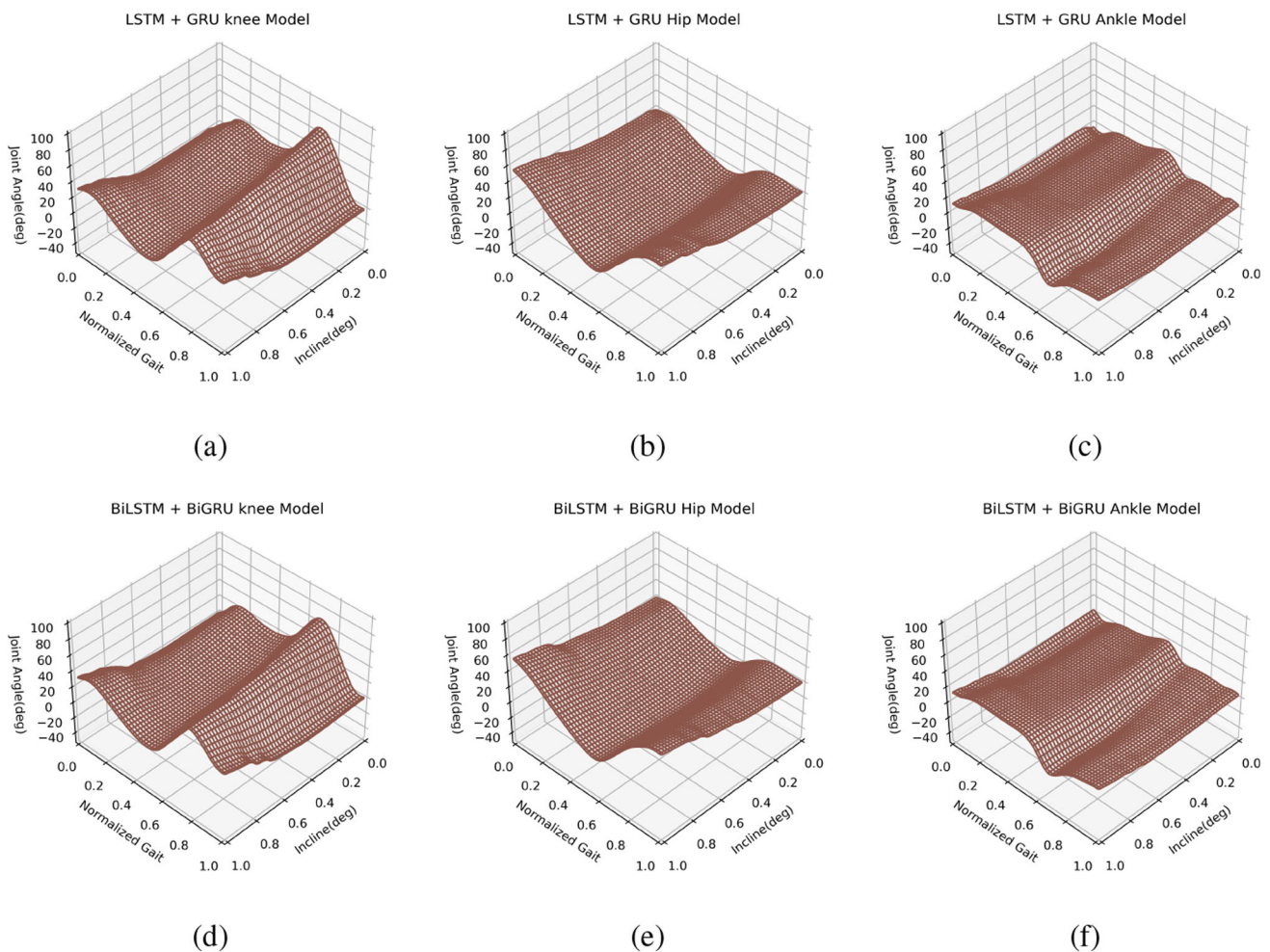


Fig. 5 Comparative analysis of data-driven models at 1 m/s speed





**Fig. 6** Comparative analysis of combine data-driven models at 1 m/s speed

combined structure of LSTM+GRU helps in capturing the changes in training joint angle trajectory data because of speed and incline variation.

## 4 Conclusion

This work presented the parametrization of the kinematic dataset using data-driven models. The dataset employed for training purposes includes the walking data of 10-able subjects on treadmill with varying speeds and inclines. The novel loss function is proposed for tuning the weights of data-driven models. The loss function incorporated the standard error of the inter-subject mean trajectory and squared difference of actual and predicted value at each point in the gait cycle. The hyper-parameter of the model is optimized using the Bayesian optimizer (given in the Keras tuner library). Also, the superiority of the proposed loss function is proved by comparing it with the other two standard loss functions i.e., mean absolute error (MAE) and mean squared error

(MSE). Afterward, the proposed model performance is validated based on the two cases. Overall, the combo of LSTM and GRU outperforms all other models in terms of statistical mean and max error indices. And, the wire-frame is also predicted for both the untrained and trained tasks over  $150 \times 150$ -time points. Finally, the impact of the varying speeds with inclines is also discussed. It is recommended to use the LSTM+GRU model for predicting the smooth joint trajectories for the biped robot or prosthetic leg.

As a future scope, authors will apply the meta-heuristic optimization approaches for the tuning of parameters of the model, and reinforcement learning will be applied to deal with the issue of non-periodic walking. Additionally, inputs such as ground slopes which are based on the external sensor can have uncertainty in their measurement. It can badly impact the predictions from the model, which is subjected to further investigation in future research.

**Author Contributions** Mr. Bharat Singh has contributed to develop the methodology, analysis, and drafting of this paper by carrying out a thorough study of the literature regarding the mentioned topics. He has

also contributed in the sense of presenting a comparison between the various techniques of gait models. All authors read and approved the final manuscript. Mr. Suchit Patel has contributed on the development of gait model. Mr. Ankit Vijayvargiya has contributed in analyzing the results and provided the suggestion in developing the gait models. Prof. Rajesh Kumar has also contributed this work by supervising all aspects related to methodology, analysis and drafting.

**Funding** The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

**Code availability** Not applicable.

## Declarations

**Conflicts of interest** All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

**Ethics approval** Not applicable.

**Consent to participate** Not applicable.

**Consent to publication** Not applicable.

## References

- Allen, D. M. (1971). Mean square error of prediction as a criterion for selecting variables. *Technometrics*, 13(3), 469–475. <https://doi.org/10.1080/00401706.1971.10488811>
- Aoi, S., & Tsuchiya, K. (2011). Generation of bipedal walking through interactions among the robot dynamics, the oscillator dynamics, and the environment: Stability characteristics of a five-link planar biped robot. *Autonomous Robots*, 30(2), 123–141.
- Burman, V., & Kumar, R. (2021). Wheel robot review. *Innovations in cyber physical systems* (pp. 781–791). Singapore: Springer.
- Canziani, A., Paszke, A., & Culurciello, E., (2016). An analysis of deep neural network models for practical applications. arXiv preprint [arXiv:1605.07678](https://arxiv.org/abs/1605.07678).
- Castillo, G. A., Weng, B., Zhang, W., & Hereid, A. (2022). Reinforcement learning-based cascade motion policy design for robust 3D bipedal locomotion. *IEEE Access*, 10, 20135–20148.
- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?-Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250. <https://doi.org/10.5194/gmd-7-1247-2014>
- Chai, L., Du, J., Liu, Q. F., & Lee, C. H. (2019). Using generalized Gaussian distributions to improve regression error modeling for deep learning-based speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12), 1919–1931. <https://doi.org/10.1109/TASLP.2019.2935803>
- Coyle, E. J., & Lin, J. H. (1988). Stack filters and the mean absolute error criterion. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(8), 1244–1254. <https://doi.org/10.1109/29.1653>
- Dey, R., & Salem, F.M., (2017). Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, IEEE (pp. 1597–1600).
- Doerschuk, P. I., Nguyen, V., & Li, A. (2002). A scalable intelligent takeoff controller for a simulated running jointed leg. *Applied Intelligence*, 17(1), 75–87. <https://doi.org/10.1023/A:1015731916039>
- Dong, Y., (2018). A survey on neural network-based summarization methods. [arXiv:1804.04589](https://arxiv.org/abs/1804.04589).
- Embry, K.R., Villarreal, D.J., Gregg, R.D., (2016). A unified parameterization of human gait across ambulation modes. In *38th annual international conference of the IEEE engineering in medicine and biology society, IEEE* (pp. 2179–2183).
- Embry, K., Villarreal, D., Macaluso, R., & Gregg, R. (2018). The effect of walking incline and speed on human leg kinematics, kinetics, and EMG. *IEEE Dataport*. <https://doi.org/10.21227/gk32-e868>
- Embry, K. R., Villarreal, D. J., Macaluso, R. L., & Gregg, R. D. (2018). Modeling the kinematics of human locomotion over continuously varying speeds and inclines. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(12), 2342–2350. <https://doi.org/10.1109/TNSRE.2018.2879570>
- Gao, Y., Wei, W., Wang, X., Li, Y., Wang, D., & Yu, Q. (2021). Feasibility, planning and control of ground-wall transition for a suctorial hexapod robot. *Applied Intelligence*, 51, 1–19. <https://doi.org/10.1007/s10489-020-01955-2>
- Gouk, H., Frank, E., Pfahringer, B., & Cree, M. J. (2021). Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2), 393–416. <https://doi.org/10.1007/s10994-020-05929-w>
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
- Hildebrandt, A. C., Schwerd, S., Wittmann, R., Wahrmann, D., Sygulla, F., Seiwald, P., & Buschmann, T. (2019). Kinematic optimization for bipedal robots: A framework for real-time collision avoidance. *Autonomous Robots*, 43(5), 1187–1205.
- Holgate, M A., Sugar, T G., Bohler, A.W., (2009). A novel control algorithm for wearable robotics using phase plane invariants. In *2009 IEEE international conference on robotics and automation, IEEE*, pp 3845–3850.
- Hosseinmemar, A., Baltes, J., Anderson, J., Lau, M. C., Lun, C. F., & Wang, Z. (2019). Closed-loop push recovery for inexpensive humanoid robots. *Applied Intelligence*, 49(11), 3801–3814. <https://doi.org/10.1007/s10489-019-01446-z>
- Huang, Z., Xu, W., & Yu, K., (2015). Bidirectional LSTM-CRF models for sequence tagging. [arXiv:1508.01991](https://arxiv.org/abs/1508.01991).
- Huang, C. F., & Yeh, T. J. (2019). Anti slip balancing control for wheeled inverted pendulum vehicles. *IEEE Transactions on Control Systems Technology*, 28(3), 1042–1049. <https://doi.org/10.1109/TCST.2019.2891522>
- Huang, B., Xiong, C., Chen, W., Liang, J., Sun, B. Y., & Gong, X. (2021). Common kinematic synergies of various human locomotor behaviours. *Royal Society open science*, 8(4), 210161. <https://doi.org/10.1098/rsos.210161>
- Juang, C. F., & Yeh, Y. T. (2017). Multiobjective evolution of biped robot gaits using advanced continuous ant-colony optimized recurrent neural networks. *IEEE Transactions on Cybernetics*, 48(6), 1910–1922. <https://doi.org/10.1109/TCYB.2017.2718037>
- Kim, J. H., Lee, J., & Oh, Y. (2018). A theoretical framework for stability regions for standing balance of humanoids based on their LIPM treatment. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(11), 4569–4586. <https://doi.org/10.1109/TSMC.2018.2855190>
- Kim, D., Jorgensen, S. J., Lee, J., Ahn, J., Luo, J., & Sentis, L. (2020). Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control. *The International Journal of Robotics Research*, 39(8), 936–956. <https://doi.org/10.1177/0278364920918014>
- Kim, N., Jeong, B., & Park, K. (2021). A novel methodology to explore the periodic gait of a biped walker under uncertainty using a machine learning algorithm. *Robotica*. <https://doi.org/10.1017/S0263574721000424>

- Kim, K., Spieler, P., Lupu, E. S., Ramezani, A., & Chung, S. J. (2021). A bipedal walking robot that can fly, slackline, and skateboard. *Science Robotics*, 6(59), eabf8136. <https://doi.org/10.1126/scirobotics.abf8136>
- Kormushev, P., Ugurlu, B., Caldwell, D. G., & Tsagarakis, N. G. (2019). Learning to exploit passive compliance for energy-efficient gait generation on a compliant humanoid. *Autonomous Robots*, 43(1), 79–95.
- Kuffner, J. J., Kagami, S., Nishiwaki, K., Inaba, M., & Inoue, H. (2002). Dynamically-stable motion planning for humanoid robots. *Autonomous robots*, 12(1), 105–118.
- Kwon, O., & Park, J. H. (2009). Asymmetric trajectory generation and impedance control for running of biped robots. *Autonomous Robots*, 26(1), 47–78.
- Lee, Y., Hwang, S., & Park, J. (2016). Balancing of humanoid robot using contact force/moment control by task-oriented whole body control framework. *Autonomous Robots*, 40(3), 457–472.
- Li, T. H. S., Su, Y. T., Lai, S. W., & Hu, J. J. (2010). Walking motion generation, synthesis, and control for biped robot by using PGRL, LPI, and fuzzy logic. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(3), 736–748. <https://doi.org/10.1109/TSMCB.2010.2089978>
- Li, Z., Tsagarakis, N. G., & Caldwell, D. G. (2013). Walking pattern generation for a humanoid robot with compliant joints. *Autonomous Robots*, 35(1), 1–14.
- Li, T. H. S., Kuo, P. H., Cheng, C. H., Hung, C. C., Luan, P. C., & Chang, C. H. (2020). Sequential sensor fusion-based real-time LSTM gait pattern controller for biped robot. *IEEE Sensors Journal*, 21(2), 2241–2255. <https://doi.org/10.1109/JSEN.2020.3016968>
- Liu, C., Zhang, T., Liu, M., & Chen, Q. (2020). Active balance control of humanoid locomotion based on foot position compensation. *Journal of Bionic Engineering*, 17(1), 134–147.
- Liu, J., Chen, H., Wensing, P. M., & Zhang, W. (2021). Instantaneous capture input for balancing the variable height inverted pendulum. *IEEE Robotics and Automation Letters*, 6(4), 7421–7428. <https://doi.org/10.1109/LRA.2021.3097074>
- Macaluso, R., Embry, K., Villarreal, D. J., & Gregg, R. D. (2021). Parameterizing human locomotion across quasi-random treadmill perturbations and inclines. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29, 508–516. <https://doi.org/10.1109/TNSRE.2021.3057877>
- Mangasarian, O. L., & Shiao, T. H. (1987). Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems. *SIAM Journal on Control and Optimization*, 25(3), 583–595. <https://doi.org/10.1137/0325033>
- Morris, K. J., Samonin, V., Baltes, J., Anderson, J., & Lau, M. C. (2019). A robust interactive entertainment robot for robot magic performances. *Applied Intelligence*, 49(11), 3834–3844. <https://doi.org/10.1007/s10489-019-01565-7>
- Pon, MZ, KK, KP., (2021). Hyperparameter tuning of deep learning models in keras. *Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (STAIQC)*. 1(1):36-40. <https://doi.org/10.55011/staiqc.2021.1104>
- Qi, J., Du, J., Siniscalchi, S. M., Ma, X., & Lee, C. H. (2020). On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*, 27, 1485–1489. <https://doi.org/10.1109/LSP.2020.3016837>
- Quintero, D., Villarreal, D. J., Lambert, D. J., Kapp, S., & Gregg, R. D. (2018). Continuous-phase control of a powered knee-ankle prosthesis: Amputee experiments across speeds and inclines. *IEEE Transactions on Robotics*, 34(3), 686–701. <https://doi.org/10.1109/TRO.2018.2794536>
- Razavi, H., Faraji, S., & Ijspeert, A. (2019). From standing balance to walking: A single control structure for a continuum of gaits. *The International Journal of Robotics Research*, 38(14), 1695–1716. <https://doi.org/10.1177/0278364919875205>
- Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, 7, 53040–53065. <https://doi.org/10.1109/ACCESS.2019.2912200>
- Siekman, J., Godse, Y., Fern, A., & Hurst, J., (2021). Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 7309–7315.
- Simon, A. M., Ingraham, K. A., Fey, N. P., Finucane, S. B., Lipschutz, R. D., Young, A. J., & Hargrove, L. J. (2014). Configuring a powered knee and ankle prosthesis for transfemoral amputees within five specific ambulation modes. *PLoS One*, 9(6), e99387. <https://doi.org/10.1371/journal.pone.0099387>
- Singh, B., Gupta, V., & Kumar, R. (2021). Probabilistic modeling of human locomotion for biped robot trajectory generation. In *2021 IEEE 8th Uttar Pradesh section international conference on electrical electronics and computer engineering (UPCON)* (pp. 1–6).
- Singh, B., Kumar, R., Singh, V. P., et al. (2021). Reinforcement learning in robotic applications: A comprehensive survey. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-021-09997-9>
- Singh, B., Vijayvargiya, A., & Kumar, R. (2022). Kinematic modeling for biped robot gait trajectory using machine learning techniques. *Journal of Bionic Engineering*, 19, 355–369. <https://doi.org/10.1007/s42235-021-00142-4>
- Sun, Y., Tang, H., Tang, Y., Zheng, J., Dong, D., Chen, X., & Luo, J. (2021). Review of recent progress in robotic knee prosthesis related techniques: Structure, actuation and control. *Journal of Bionic Engineering*, 18(4), 764–785.
- Vicon Motion Systems Ltd. (2018) Vicon nexus reference guide. <https://docs.vicon.com/display/Nexus26/Vicon+Nexus+Reference+Guide>
- Wang, L., Liu, Z., Chen, C. P., & Zhang, Y. (2014). Interval type-2 fuzzy weighted support vector machine learning for energy efficient biped walking. *Applied Intelligence*, 40(3), 453–463. <https://doi.org/10.1007/s10489-013-0472-2>
- Wang, Z., He, B., Zhou, Y., Yuan, T., Xu, S., & Shao, M. (2018). An experimental analysis of stability in human walking. *Journal of Bionic Engineering*, 15(5), 827–838.
- Wang, K., Ren, L., Qian, Z., Liu, J., Geng, T., & Ren, L. (2021). Development of a 3D printed bipedal robot: Towards humanoid research platform to study human musculoskeletal biomechanics. *Journal of Bionic Engineering*, 18(1), 150–170.
- Yamaguchi, J. I., & Takanishi, A. (1997). Development of a leg part of a humanoid robot-development of a biped walking robot adapting to the humans' normal living floor. *Autonomous Robots*, 4(4), 369–385.
- Yang, W., Tan, R. T., Wang, S., Fang, Y., & Liu, J. (2021). Single image deraining: From model-based to data-driven and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11), 4059–4077. <https://doi.org/10.1109/TPAMI.2020.2995190>
- Zhou, C., Wang, X., Li, Z., & Tsagarakis, N. (2017). Overview of gait synthesis for the humanoid COMAN. *Journal of Bionic Engineering*, 14(1), 15–25.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.





**Bharat Singh** (Student Member, IEEE) received the B.Tech. (Hons.) degree in electrical engineering from RTU Kota, India, in 2013, the M.Tech. degree in integrated power system from VNIT, Nagpur, India, in 2016. He is currently pursuing a Ph.D. degree in electrical engineering from the MNIT, Jaipur, India. His research interests include Robotics & Control, Reinforcement learning, and machine learning.



**Ankit Vijayvargiya** (Student Member, IEEE) received the B.Tech. (Hons.) degree in electrical engineering from RTU Kota, India, in 2010, the M.Tech. degree in instrumentation from Devi Ahilya University, Indore, India, in 2014. He is currently pursuing a Ph.D. degree in electrical engineering from the MNIT, Jaipur, India. His research interests include biomedical signal processing and machine learning.



**Suchit Patel** received his integrated B.Tech-M.Tech degree from UoR Jaipur, India, in 2022. His research interests are deep learning, machine learning, and robotics.



**Rajesh Kumar** (Senior Member, IEEE) received the B.Tech. (Hons.) degree in electrical engineering from the NIT Kurukshetra, India, in 1994, the M.E. (Hons.) degree in power systems from the MNIT, Jaipur, Rajasthan, India, in 1997, and the Ph.D. degree in intelligent systems from the Our Jaipur, India, in 2005. Since 1995, he has been a Faculty Member of the Department of Electrical Engineering, MNIT, where he is currently associated as a Professor. He was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore, from 2009 to 2011. His research interests include intelligent systems, computational intelligence, and robotics.