DeepBBWAE-Net: A CNN-RNN Based Deep SuperLearner For Estimating Lower Extremity Sagittal Plane Joint Kinematics Using Shoe-Mounted IMU Sensors In Daily Living

SUBMISSION DATE / POSTED DATE

23-07-2021 / 11-04-2022

CITATION

Hossain, Md Sanzid Bin; Dranetz, Joseph; Choi, Hwan; Guo, Zhishan (2021): DeepBBWAE-Net: A CNN-RNN Based Deep SuperLearner For Estimating Lower Extremity Sagittal Plane Joint Kinematics Using Shoe-Mounted IMU Sensors In Daily Living. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.15040653.v2

DOI

10.36227/techrxiv.15040653.v2

DeepBBWAE-Net: A CNN-RNN Based Deep SuperLearner For Estimating Lower Extremity Sagittal Plane Joint Kinematics Using Shoe-Mounted IMU Sensors In Daily Living

Md Sanzid Bin Hossain, *Student Member, IEEE*, Joseph Dranetz, *Student Member, IEEE*, Hwan Choi, *Member, IEEE*, Zhishan Guo, and *Senior Member, IEEE*

***Abstract*— Measurement of human body movement is an essential step in biomechanical analysis. The current standard for human motion capture systems uses infrared cameras to track reflective markers placed on a subject. While these systems can accurately track joint kinematics, the analyses are spatially limited to the lab environment. Though Inertial Measurement Units (IMUs) can eliminate these spatial limitations, those systems are impractical for use in daily living due to the need for many sensors, typically one per body segment.**

Due to the need for practical and accurate estimation of joint kinematics, this study implements a reduced number of IMU sensors and employs a machine learning algorithm to map sensor data to joint angles. Our developed algorithm estimates hip, knee, and ankle angles in the sagittal plane using two shoe-mounted IMU sensors in different practical walking conditions: treadmill, overground, stair, and slope conditions. Specifically, we proposed five deep learning networks that use combinations of Convolutional Neural Networks (CNN) and Gated Recurrent Unit (GRU) based Recurrent Neural Networks (RNN) as base learners for our framework. Using those five baseline models, we proposed a novel framework, DeepBBWAE-Net, that implements ensemble techniques such as bagging, boosting, and weighted averaging to improve kinematic predictions. DeepBBWAE-Net predicts joint kinematics for the three joint angles for each of the walking conditions with a Root Mean Square Error (RMSE) 6.93-29.0% lower than the base models individually. This is the first study that uses a reduced number of IMU sensors to estimate kinematics in multiple walking environments.

***Index Terms*— Kinematics Estimation; Wearable IMU Sensors; Ensemble Learning; Deep Learning; Machine Learning**

I. Introduction

Human gait is the complex and coordinated movement of the limbs which allows for locomotion. Gait motion analysis is a qualitative and quantitative tool used to assess people's kinesiological health. For example, as people age, they experience transformations in their gait pattern [1], [2]. Early detection of this transformation is useful in assessing fall risk among the elderly [2]. Gait abnormalities can also occur due to neurological damage related to traumatic injury or disease [3]. As gait alterations can significantly impact quality of life, gait analysis has become a more popular and relevant research topic. Gait motion assessment can aid in the evaluation of the severity, progression, and diagnosis of a disease or injury. Joint kinematics are necessary spatiotemporal measures to assess abnormal gait function in the clinical setting. For example, knee joint kinematics have been used to evaluate hypermobility syndrome (HMS) in children [4]. In addition to disease assessment, gait motion analysis is also a valuable tool in the field of sports medicine [5]. The ability to evaluate gait without a traditional lab setup facilitates patient monitoring, accelerating the rehabilitation process. In [6], the authors used a set of wearable sensors to track human lower limb motion, presenting a method for measuring trajectories of the lower limb without an optical motion capture system. In [7], a SmartShoe was developed to monitor the physical activity and gait patterns of children with CP in community living. In [8], a wearable biofeedback suit was developed to estimate joint kinematics during aquatic exercises to provide biofeedback to therapists while patients are underwater.

Traditional movement measurement techniques for gait assessment use 3D infrared light motion capture cameras to measure the trajectories of reflective markers placed on the human body. The acquired data is then processed with dynamic analysis software to compute joint angles [OpenSim [9], Visual3D (C-Motion, USA), Vicon Nexus (Oxford, UK)]. Although this method is regarded as the ground truth measure for dynamic function of human motion [10], it requires manual data processing, expert device operation and only works in a spatially and temporally restricted area due to the need for a large number of cameras to define a limited capture volume. It has been suggested that the use of Inertial Measurement Unit (IMU) sensors may overcome some of the disadvantages with current methods for assessing the dynamic function of human motion. Recent studies have shown that IMU's can be used to calculate joint angles [11]. Researchers can estimate human movements outside of the laboratory environment by replacing traditional motion capture cameras with wearable IMU sensors. These techniques also enable potential applications in long-term use and integrative daily monitoring. However, the use of IMU's requires setup before collecting motion data. For example, each IMU sensor needs to be calibrated for each specific body segment as the position and orientation of every sensor assigned to each segment must be defined. While several calibration processes have been proposed [12], [13], they are prone to error as they are challenging to implement onto subjects consistently. Traditional IMU-based estimation also requires the placement of sensors onto each segment of the body, making them too cumbersome for application in daily/continuous monitoring. Mundt et al. found that significant redundancy can occur in a multi-sensor human motion capture system such that it is possible to reduce the number of sensors without a significant loss in kinematic accuracy [14].

Challenges. The key challenge to estimating joint angles using a reduced number of sensors is the lack of input signal. Human movement is complex in nature [15], and it is challenging to infer several joint angles using only sensors at the feet. However, it is possible to infer the missing IMU signals virtually by utilizing statistic methods and signal processing techniques [16]–[18] while using a reduced number of sensors. These statistical methods and signal processing techniques are resource-intensive and impractical for application in real-time gait monitoring. A machine learning

Md Sanzid Bin Hossain is with the Electrical and Computer Engineering Department, University of Central Florida, Orlando, FL 32816, USA (e-mail: sanzid@knights.ucf.edu)

Joseph Dranetz is with the Mechanical and Aerospace Engineering Department, University of Central Florida, Orlando, FL 32816, USA (e-mail: jmdranetz@knights.ucf.edu)

Hwan Choi is with the Mechanical and Aerospace Engineering Department, University of Central Florida, Orlando, FL 32816, USA (e-mail: hwan.choi@ucf.edu)

Zhishan Guo is with the Electrical and Computer Engineering Department, University of Central Florida, Orlando, FL 32816, USA (e-mail: zsguo@knights.ucf.edu)

model may replace these statistical methods and signal processing techniques to allow for real-time estimations of kinematics.

As of late, there has been a massive wave of innovations in machine learning. A properly-trained model can estimate kinematics with decent accuracy [19]–[22]. Machine learning models implementing Gated Recurrent Unit (GRU) based Recurrent Neural Networks (RNN), or Convolutional Neural Networks (CNN) can extract relevant features from the raw data of a reduced set of IMU sensors to estimate kinematics. As a result, these machine learning models can help overcome the limitations introduced by static pose calibration, biomechanical modeling complexity, magnetometer data error, and multi-sensor constraints.

Ensemble learning is a classical machine learning approach that has created significant attention among researchers in recent decades. The attention is well deserved as it yields significant performance improvements in many real-world applications. It is the technique of combining multiple weak learners in an intelligent way to create a stronger learner, which has better predictive performance than those of single weak learners individually. There are many well-established ensemble learning methods: bagging, boosting, stack generalization, weighted average ensemble, etc. In some of these methods, training data is manipulated, passed to a homogeneous learner, and then combined using averaging or majority voting for regression and classification problems, respectively. In other methods, parameters or the structures of the models are varied to create multiple learners and then combined using either a secondary learner called a meta-learner or a weighted averaging system to get the final ensemble model's prediction. To get accurate kinematic estimations in daily living, a combination of different ensemble approaches may be helpful.

Contribution. This work aims to estimate kinematics (sagittal plane hip, knee, and ankle joint angles) under various walking conditions (stair, slope, overground, and treadmill) using just two IMU sensors (one per shoe). We propose five deep learning models consisting of Convolutional Neural Networks (CNN) and Gated Recurrent Unit (GRU) based Recurrent Neural Networks (RNN) for time-series data predictions. These models are well suited to generate features from raw IMU data to make predictions. We further develop a novel CNN-RNN based deep super learner model, Deep Bagging, Boosting, and Weighted Average Ensemble (DeepBBWAE-Net). This super learner implements various traditional ensemble methods (bagging, boosting, weighted average ensemble) to combine the deep learning neural networks. Combining predictions from multiple models improves the accuracy of the joint angle estimation. Our experimental results suggest that the proposed super learner method outperforms each of the learners individually. Furthermore, DeepBBWAE-Net uses raw data as the input of the neural network without any feature extraction or time normalization and minimizes the system's time complexity, paving the way for real-time gait monitoring.

The rest of this paper is organized as follows: Section II briefly reviews recent literature about machine learning-based kinematics estimation using IMUs and the fundamental machine learning concepts that are used in the construction of the DeepBBWAE-Net framework. The problem statement as well as the structures of the DeepBBWAE-Net framework and its base learners are discussed in Section III. Section IV describes the procedures followed for data collection, data pre-processing, the parameters chosen for the machine learning model, statistical validation methods, and the results of the prediction. Section V discusses implications of these results, describes study limitations, and proposes future work. We conclude the paper in Section VI.

II. Related Work

This section will discuss the current state-of-the-art machine learning methods for IMU-based kinematics estimation and traditional machine learning components used in our framework. First, we will discuss studies where a complete set of IMU sensors was used to estimate the kinematics. Then we will discuss attempts to employ a reduced number of IMU sensors to achieve similar kinematic predictions. After that, we will discuss the literature related to traditional machine learning techniques used in our framework.

A. IMU and Machine learning-based approach

Mundt et al. [19] used five IMU sensors to estimate 3D joint kinematics. They segmented and normalized their time-series data into gait cycles before inputting it into a feed-forward neural network. By enlarging the training data set with simulated IMU signals using marker trajectory data, they decreased the prediction's mean RMSE from 4.8° to 4.3° and increased its correlation coefficient from 0.85 to 0.89. In [14], [23], the author used simulated IMU sensor data for 3D kinematic prediction using a Long Short Term Memory (LSTM) and Artificial Neural Network (ANN) based model to achieve correlation coefficients higher than 0.98. However, because they only validated their study with simulated data, it is difficult to determine how their approach will perform with actual IMU signal. In [20], a simulation technique with a musculoskeletal model was used to augment a data set to train a deep learning model using a 2D convolutional layer to predict kinematics for running and walking using four IMU sensors on a single leg. Their simulation technique was able to decrease the RMSE of hip, knee, and ankle joint angles by 17%, 27% and 23% respectively. In [24], five IMU sensors were used to estimate kinematics in walking and running on a treadmill using a deep model based on convolutional and long-short term memory recurrent layers (DeepConvLSTM) with mean absolute errors ranging from 2.2° to 5.1°. More recently, in [25], the authors combined deep learning and optimization frameworks to estimate 3D kinematics using seven IMU sensors for walking and running, achieving an RMSE of 1.27° (± 0.38°) for knee flexion/extension. However, because they only validated their study with simulated IMU data, it is unclear how their method will perform using real IMU data. These studies have used a relatively large set of sensors and were only conducted on treadmill or overground conditions.

B. Reduced Sensor-Based Approach

Many studies have employed machine learning for kinematics estimation of treadmill and overground walking conditions [19], [14], [23], [24], [20], [25] using a large number of sensors (4-7 sensors). Due to the impracticality associated with using a full set of IMU sensors, some studies have employed a reduced set of IMU sensors to estimate joint kinematics. Lim et al. [21] used a single IMU mounted on participants' sacrum to estimate kinetics and kinematics, achieving RMSEs of 3.1°, 2.2°, 3.4° for hip, knee, and ankle angles during the stance phase. To do this, they processed the IMU data to calculate acceleration, velocity, and displacement over time and used this data as input for an ANN. This data processing was done retroactively, and the pre-processing they performed may have resulted in an additional error. The method was also not able to estimate kinetics and kinematics for both legs simultaneously. Gholami et al. [22] proposed a single accelerometer based approach for calculating the kinematics of running on a treadmill and achieved RMSEs of 5.6°, 6.5°, 4.7° for hip, knee, and ankle angles with an inter-participant model (exclusion of test subject during training). As their deep learning model uses input data from past and future samples for present kinematics prediction, it cannot make predictions for the start and the end portions of the data. As they have built their model based on treadmill data, a repetitive process, the predictions are good as it is easier to predict repetitive kinematics when both past and future

data are provided. But in an uncontrolled environment like stair and slope conditions, their predictive model may not perform well. More recently, Alcaraz et al. [26] used a single IMU sensor on the foot to estimate kinematics during overground walking. They have achieved an average RMSE of 1.91°, 2.12° and 2.57° for the hip, knee, and ankle joints. The reason for their good prediction may be due to them using 80/20% splits of each participant's data for training/testing sets, resulting in same subject's data in both sets. Additionally, they used gait cycle normalization, a Hilbert-Huang transformation, to process the data making it difficult for real-time prediction due to the longer computation time. Moreover, using a single sensor on the foot may create ambiguity when the foot is in full contact with the ground because the IMU doesn't record any acceleration or angular velocity, but the hip, knee, and ankle angles are still changing.

C. Fundamental Neural Network Components and Methodologies

This study aims to use machine learning techniques to estimate lower extremity joint angles as accurately as possible. Using an effective machine learning model is an essential step for this purpose. In this subsection, we will briefly discuss the theoretical details of the methods used to build DeepBBWAE-Net and justify their use.

Gated Recurrent Unit (GRU). Long-short term memory (LSTM) is a special type of Recurrent neural network (RNN) that can capture time-dependency in time series data without suffering from the gradient vanishing or explosion problem [27], and thus can process longer sequences of data (compared to regular RNN). Nevertheless, LSTM is heavily parameterized and takes a longer time to train. Cho et al. [28] first proposed Gated Recurrent Units (GRU), a special case of LSTM without an output gate, resulting in fewer parameters and a reduction in the required training time. GRU will be used for feature learning from the accelerometer and gyroscope time-series data in our study.

Convolutional Neural Networks (CNN). CNN's are computationally optimized feedforward deep neural networks with sparse convolutional layers. Each convolutional layer has kernel functions that convolute small patches of the input signal. This makes convolutional networks more optimized for feature extraction and less computationally expensive. The use of pre-processed data from the convolutional network is advantageous as handcrafted features extracted by humans may not sufficiently represent the underlying relations. CNN models can use raw data from the input to learn their feature representation through backpropagation while maximizing the best mapping of the IMU data to the joint kinematic data.

Bagging. Bootstrap aggregating (bagging) is a classic ensemble machine learning technique proposed by Breiman [29] which can be applied in both classification and regression problems. Bootstrap samples are created using random sampling with replacement from a data set. Each bootstrap data subset is then passed to a homogeneous learner to create multiple distinct trees, which can be aggregated by voting or averaging for classification and regression problems, respectively. Bagging reduces the variance of prediction and increases the stability and accuracy of the machine learning model.

Boosting. In bagging, multiple models are trained independently, while in boosting, multiple weak learners are trained sequentially in an adaptive way to form a stronger learner. To reduce bias during the learning process, models adaptively give more weight to the instances in the data set that were weakly interpreted by previous models (and less weight to cases of successful interpretations). We implement Gradient Boosting Machine (GBM) [30], which can be considered an optimization process toward finding a model that minimizes loss.

Weighted Average Ensemble (WAE). Model averaging is another ensemble approach where predictions from each model are aggregated to generate a final prediction in a regression problem. Different weights are applied to different models according to their effectiveness toward the prediction to ensure optimal performance from the final ensemble model.

Convex Optimization. Sequential least-square programming (SLSQP) is a gradient-based optimization method, which was first proposed by Kraft and Schnepper [31]. In our framework, we use this optimization procedure to find the optimal weights of the models and to minimize the RMSE between ground truth and the predictions.

III. Proposed Approach

In this section, we will discuss the problem statement, detailed workflow of DeepBBWAE-Net Framework, and base learners used for building the framework.

A. Problem Statement

The purpose of the study is to estimate joint angles of the hip, knee, and ankle using two shoe-mounted sensors in different practical walking scenarios: stair, slope, and overground. To date, no study has attempted to employ a reduced number of sensors to estimate joint angles for various walking conditions. As the number of sensors is reduced, an effective machine learning model is essential for accurate kinematic estimations. For this reason, we proposed a novel framework, DeepBBWAE-Net, which leverages bagging, boosting, and weighted average ensemble techniques to improve the prediction performance. We believe two sensors should be the minimum necessary number and thus set this limit in our design. When one foot is in full contact with the ground, its IMU signal becomes zero, so a sensor on the other foot is necessary to provide the network with information about kinematics at that time.

B. DeepBBWAE-Net Framework

The DeepBBWAE-Net framework consists of three fundamental blocks: BagBoost, Weighted Average Ensemble (WAE), and BaseBoost (Fig. 1). The BagBoost block consists of multiple BagBoost cells. The function of each BagBoost cell is to apply bagging and boosting to each base learner to improve the prediction performance. The function of the BaseBoost block is to generate boosted predictions of the base learner using the whole data set. This block gives additional information to the framework for improving the prediction performance. The function of the WAE block is to combine the outputs of the multiple BagBoost and BaseBoost blocks using optimal weighted averaging to generate the final prediction.

BagBoost block. The BagBoost block consists of multiple BagBoost cells. The number of BagBoost cells (n) will depend on the number of base learners used in the framework. Fig. 2 demonstrates the workflow of a single BagBoost cell. In each cell, a single base learner is used to get two predictions: Y_{bag} and $Y_{bagboost}$. Ensemble learning–bagging and GBM are used to get these two predictions. At first, k bootstrap samples are created from the training data, and each sample is used to train the same base learner. Predictions from each base learner are then averaged to get Y_{bag}. Additionally, each Y_{bag} is passed to a GBM, trained with the validation data, to increase generalization. The prediction from each GBM learner is then averaged to get $Y_{bagboost}$.

BaseBoost block. Unlike in the BagBoost cells, the entire training data set is used to train the base learners in the BaseBoost block. To further improve the performance, results from the base learners are passed to a GBM, trained with the validation data, and has an output named $Y_{baseboost}$.

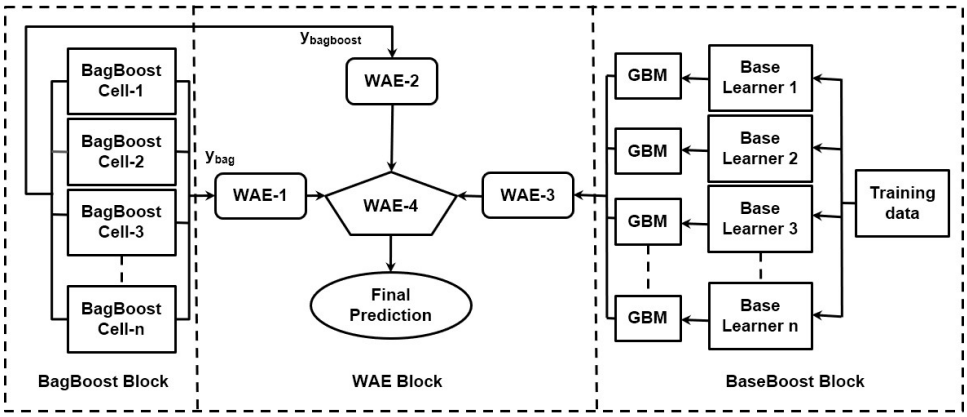

Fig. 1. DeepBBWAE-Net consisting of three blocks- a BagBoost, WAE, and BaseBoost Block

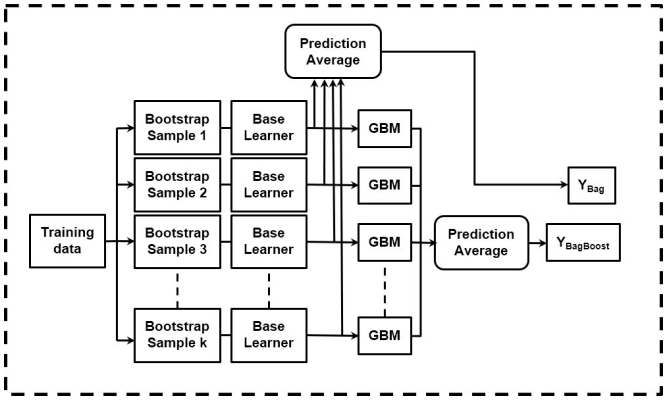

Fig. 2. A BagBoost Cell

WAE block. In the WAE block, we have four WAE cells: WAE-1, WAE-2, WAE-3, WAE-4. The function of each cell is to combine results from multiple branches to generate a better prediction.

In WAE-1, prediction Y_{bag} from the BagBoost cells are used to generate Bag-WAE-Net. In WAE-2, prediction $Y_{bagboost}$ from the BagBoost cells are used to create BagBoost-WAE-Net. In WAE-3, prediction $Y_{baseboost}$ from the BaseBoost cells are used to generate BaseBoost-WAE-Net. WAE-1, WAE-2, and WAE-3 are then used to build WAE-4, which produces the final prediction. To find the optimal weights in the WAE block, we used a convex optimization method, the SLSQP optimization algorithm, where the loss function was defined as the RMSE between ground truth kinematic data and the kinematic predictions.

C. Base learners

This subsection will discuss the details of the five base learners that we have developed to build DeepBBWAE-Net. We used several blocks consisting of 1D/2D convolutional layers, Bi-GRU layers, and fully connected layers to create the base learners. First, we will introduce the base learners. Then, we will describe the different fundamental blocks used to create the base learners.

1) Base learner 1: Conv2D-Net: In base learner 1 (Fig. 3), two fundamental blocks, i.e., 2D convolutional and fully connected blocks, were used. First, batch normalization (BN) is applied on the input signal to address the heterogeneity of source data [32] to perform regularization. After batch normalization, a 2D convolutional block and a fully connected block were added consecutively. A flattening layer is added to flatten the output from the fully connected blocks, which is then connected to the last prediction layer.

2) Base learner 2: Bi-GRU-Net: In Base learner 2 (Fig. 3), a Bi-GRU block is used after batch normalizing the input data. A flattening layer was then added to flatten the output from Bi-GRU block and connected to the final output layer.

3) Base learner 3: Hybrid Conv1D-GRU-Net: Base learner 3 (Fig. 3) consists of a Bi-GRU, a 1D convolutional, and a fully connected block. After batch normalization, features are extracted from the input using the Bi-GRU block. The output of the Bi-GRU block is then passed to a 1D convolutional block to extract the features further. A fully connected block was added after the 1D convolutional block. The output from the fully connected block was then flattened to connect to the output layer.

4) Base learner 4: Hybrid Conv2D-GRU-Net-1: In base learner 4 (Fig. 3), we concatenated the output from the flattening layer of base learners 1 and 2. Both Conv2D-Net and Bi-GRU-Net were used to extract features from the IMU data. To get advantages from both model's feature extraction, we concatenated features from both of the models. This concatenated layer is then connected to the output layer for prediction.

5) Base learner 5: Hybrid Conv2D-GRU-Net-2: In base learner 5 (Fig. 3), a 2D convolutional block is first used to extract features. A regular flattening layer will flatten the input into a 2D tensor and reduces the temporal and feature dimensions into a single one. For this reason, we instead used a TimeDistributed wrapper of Keras [33] to flatten the output to produce a 3D tensor, which is the required input of the Bi-GRU layer. With this, a Bi-GRU block was added after the 2D convolutional block to extract features further. Finally, we flattened the output and connected it to a dense layer for predicting the joint angles.

6) Fundamental Blocks: This subsection will discuss the fundamental blocks specifically designed to create effective base learners for our problem. Fig. 4 demonstrates all the fundamental blocks used to build the base learners.

2D Convolutional Block. A 2D Convolutional Block consists of 2D convolutional, batch normalization (BN), and max-pooling layers. First, we have a 2D convolutional layer followed by a batch normalization layer. This batch normalization helps reduce the internal covariance shift of the network [34]. Internal covariance shift is the change in data distribution during model training as weights of the layers change during the training process. A max-pooling layer is then applied to reduce the feature space, reducing the complexity of the training [35]. Max-pooling also helps to select dominant convolved features to ensure efficient model learning. Convolutional, batch normalization, and max-pooling layers form the basic component of the 2D Convolutional block. This series of components are repeated

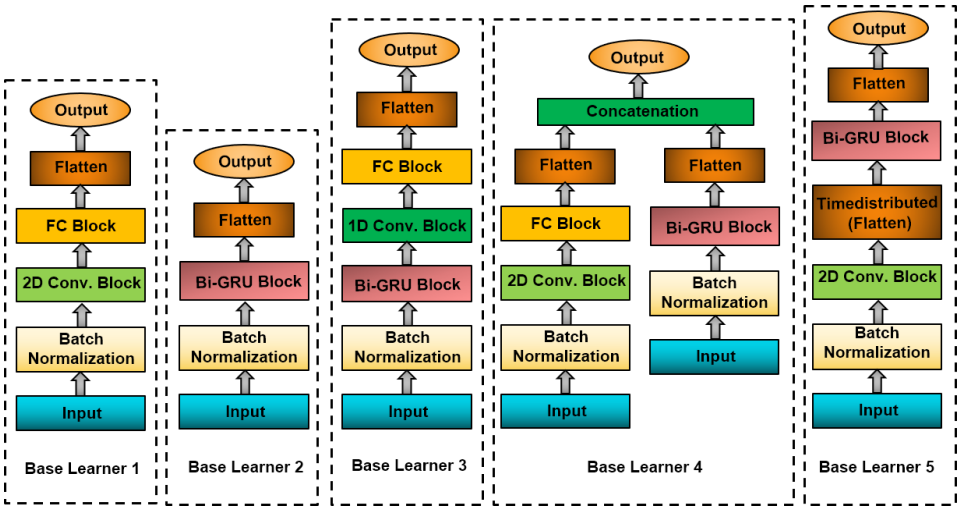

Fig. 3. All five base learners consist of: Fundamental Building Block, Input, Batch Normalization, Flatten, Concatenation, and Output layers

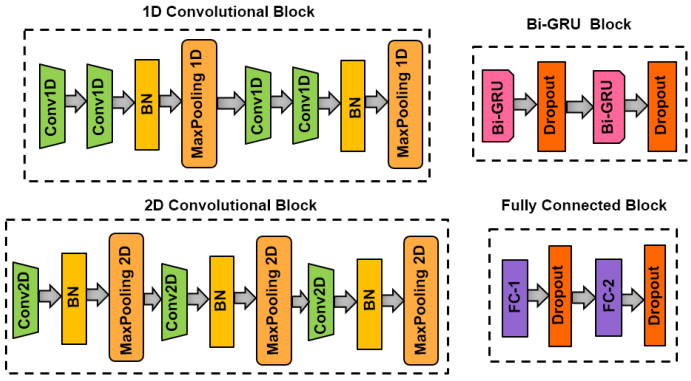

Fig. 4. The fundamental building blocks of the base learners: 1D Convolutional Block, Bi-GRU Block, 2D Convolutional Block, and Fully Connected (FC) Block

serially three times to build the 2D convolutional block.

1D Convolutional Block. In a 1D convolutional block, two convolutional layers, a batch normalization layer, and a max-pooling layer are used as a basic building block. Two of these basic building block sequences are combined in series to construct the 1D convolutional block.

Bi-GRU Block. In bidirectional GRU, two GRU layers are trained simultaneously with input windows in both the positive and negative time directions [36]. As a result, bidirectional GRU can learn feature representation from both the past and the future for its predictions. This learning gives additional context to the network for training and often results in better model performance. Dropout layers were added after the Bi-GRU layers to avoid overfitting. A series of two Bi-GRU layers, each followed by a dropout layer, is used in the Bi-GRU block.

Fully Connected Block. A fully connected block consists of two fully connected dense layers (FC-1, FC-2). To avoid overfitting, a dropout layer was added after each fully connected dense layer.

IV. Experiments

A. Data Collection

Ten healthy subjects (six male, four female, age: 23.9±2.91 years, height: 1.65±0.06 m, weight: 63.41±6.81 kg) participated in the study. Table I displays the demographics information of the participants. The informed written consent of all participants is received before participation in the experiment. The Institutional Review Board (IRB) of the University of Central Florida (UCF) approved the study's protocol (IRB ID: STUDY00002011).

Two inertial measurement unit sensors were placed onto the participants' shoes with a pre-defined sensor orientation (Fig.5). Each participant performed twelve trials: 4 treadmill, 4 overground, 2 stair, and 2 slope. Participants walked on a 5m overground at four self-selected speeds,i.e., slow, normal, fast, and very fast. Each subject performed two trials on five-step stairs (27 inch wide, 9 inch deep, and 7 inch rise steps) with their self-selected speed. Participants also performed inclined and declined walking on a slope of 20% [37], [38] at a self-selected speed. The participants walked on the treadmill at four different speeds for approximately 2 minutes. Non-dimensional slow, normal, fast, and very fast walking velocities were determined based on subject leg length, measured from the greater trochanter to the ground [39]. The actual treadmill speed was calculated by multiplying non-dimensional velocity with $\sqrt{gL_{leg}}$, where g is the gravitational acceleration, and L_{leg} is the length of the participant's leg. During the trials, thirty-four reflective markers were placed on the participant (Fig. 5) based on a modified Helen-Hayes marker set [40]. Three-dimensional marker trajectories were captured with twelve infrared light cameras (Vicon, Oxford, UK) with a sampling rate of 100 Hz. The accelerometer and gyroscope data were recorded with a sampling frequency of ∼148 Hz (sampling period: 6.75 ms) using Avanti wireless EMG/IMU (Delsys, Boston, MA). A trigger signal was sent from the motion capture system to start the left and right leg IMUs synchronously. We used OpenSim [9], an open-source musculoskeletal analysis tool, to calculate joint angles during walking conditions. First, we scaled a generic musculoskeletal model while the participant held an anatomical pose. We performed inverse kinematics after scaling to estimate kinematics of the 23 coordinates of the musculoskeletal model for the dynamic walking trials. The sagittal plane angles of the hip, knee, and ankle for both legs, six joint angle coordinates, were taken as ground truth kinematics used in the training and validation of the machine learning models.

B. Data Pre-processing

Marker data collected from the motion capture system have a frequency of 100 Hz, where IMU data were collected at a rate of ∼148 Hz. IMU data were resampled to 100 samples per second to synchronize with the motion capture data before inputting to the machine learning model. In addition to the three-axis accelerometer

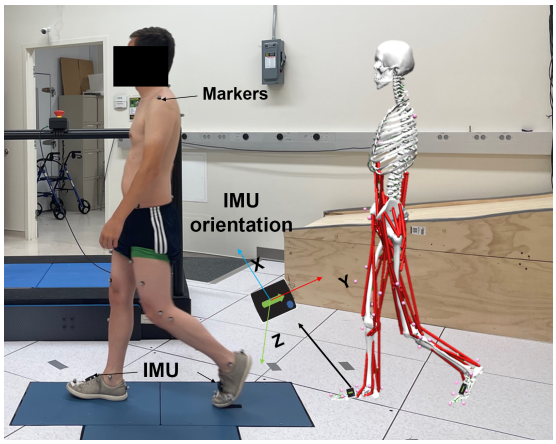

Fig. 5. Placement of markers and IMU sensors. Each IMU sensor was mounted on the dorsum of foot in the same orientation for each participant. Musculoskeletal model was created using Scale tool of OpenSim.

TABLE I
DEMOGRAPHICS OF THE PARTICIPANTS

Subject	Gender	Age	Weight (Kg)	Height (m)	Length of leg (m)
Subject 1	Male	28	60.18	1.70	0.75
Subject 2	Female	22	53.47	1.55	0.71
Subject 3	Male	21	65.54	1.73	0.73
Subject 4	Male	26	55.62	1.62	0.70
Subject 5	Male	21	72.90	1.74	0.81
Subject 6	Male	22	74.48	1.68	0.77
Subject 7	Male	26	68.90	1.67	0.70
Subject 8	Female	21	65.30	1.61	0.80
Subject 9	Female	23	58.00	1.60	0.77
Subject 10	Female	29	59.69	1.60	0.77

and gyroscope data, we also calculated the norm of three-axis acceleration and angular rotation, resulting in eight features from each IMU sensor. We segmented the data into individual gait cycles for overground, stair, and slope trials, discarding the gait initialization phase. We allowed participants to initiate gait with either leg on the stair and slope trials. This will not introduce any bias to the model as we have sensors on both left and right feet. Table II shows maximum and minimum angles for hip, knee, ankle found with motion capture for different walking conditions to put later prediction RMSEs into context. In Table III, we showed the total time and number of gait cycles for each condition.

The machine learning models considered a window length of 80 frames (0.80s) as an input. We found that 80 frames of data gave satisfactory results for joint angle prediction. As the 80 frame input includes data from 6 joint angles, we have an output size of 480 for each 80 frame time interval.

One subject's data were set aside as the testing set for leave-out-one subject cross-validation from the whole data set. The remainder of the data were split into a training set (80%) for training the base learners and a validation set (20%) to validate the model.

C. Implementation Details

We implemented our proposed algorithm in Keras on an NVIDIA Tesla P100 GPU with a training time of approximately 13 hours. The total inference time for DeepBBWAE-Net is approximately 60ms for a 0.8s input window. As DeepBBWAE-Net has multiple independent branches in its framework, the inference time can be lower down to 6ms by implementing parallel computing of ten branches of the BagBoost Cell (Fig. 2).

TABLE II
MEAN AND STANDARD DEVIATION OF MAXIMUM AND MINIMUM HIP, KNEE, AND ANKLE ANGLES FOR DIFFERENT WALKING SCENARIOS

Walking Condition		Hip (°)	Knee (°)	Ankle (°)
Treadmill	**Max**	26.66 ± 8.17	0.03 ± 3.96	13.47 ± 4.13
	Min	-22.85 ± 10.44	-65.19 ± 15.77	-18.28 ± 6.75
Overground	**Max**	22.03 ± 7.86	-1.37 ± 4.3	14.14 ± 4.92
	Min	-19.98 ± 12.89	-63.14 ± 19.32	-15.82 ± 7.78
Slope Ascent	**Max**	39.83 ± 13.55	-9.6 ± 11.86	21.01 ± 5.21
	Min	-13.63 ± 17.67	-66.38 ± 7.91	-11.18 ± 10.96
Slope Descent	**Max**	15.47 ± 4.29	-2.09 ± 4.1	22.3 ± 5.45
	Min	-11.4 ± 6.37	-76.83 ± 8.12	-9.21 ± 4.97
Stair Ascent	**Max**	55.72 ± 10.4	-12.33 ± 8.45	25.5 ± 4.78
	Min	-1.41 ± 9.57	-97.94 ± 12.07	-12.89 ± 10.22
Stair Descent	**Max**	30.09 ± 9.17	-12.77 ± 8.1	36.22 ± 8.38
	Min	0.31 ± 6.52	-93.4 ± 13.96	-23.56 ± 10.61

TABLE III
TOTAL DATA SET TRIAL TIMES FOR VARIOUS WALKING CONDITIONS

Condition	Trial Time (s)	No of Gait Cycle
Treadmill	5192.36	4621
Overground	1683.78	1458
Slope Ascent	767.48	609
Slope Descent	780.24	609
Stair Ascent	751.21	482
Stair Descent	662.12	480

1) DeepBBWAE-Net: Ten bootstrap samples (k=10) were generated for each BagBoost cell. We found that ten bootstrap samples work well for our problem. Bootstrap aggregation (bagging) is applied when the model becomes overfit, and significant variance is present in the data set. During training, the model may seem to fit the training data perfectly. When the model estimates the kinematics of a novel subject, it may not perform well due to high naturally occurring inter-subject gait pattern variability. So, indirectly the model has become overfit to the training data. As bagging is applied when the model becomes overfit and the data set has high variance, bagging can be an excellent approach to more accurate joint angle prediction. To further improve the generalization of joint angle prediction, we use GBM after bagging. Data used to validate the base models was utilized for training the gradient boosting model. Although we use dropout to avoid overfitting, the model had poor performance with the validation data compared to the training data. For this reason, to improve the generalization of the model, we used gradient boosting (GBM) techniques after the bagging prediction. We used the same methods after each base learner when training the base learners with the whole data set.

All the cells of the WAE block are generated using predictions from the leave-out-one subject cross-validation data. More specifically, we first estimated the kinematics of each subject with the BagBoost and BaseBoost blocks using leave-out-one cross-validation. We use the rest of the subjects' leave-out-one prediction data to find the optimal weights for each cell in the WAE block during test subject evaluation. We also compared the weighted average ensemble prediction of the base learners (named Base-WAE-Net) against our proposed method. We have used five base learners (n=5) to create the DeepBBWAE-Net framework.

2) Base Learners: In this subsection, we will provide details on the implementation of the base learners.

Conv2D-Net. As a 2D convolutional layer requires a 4D tensor, we reshaped the 16 inputs of two sensors into 8 $features \times 2$ $sensors$. For three convolutional layers in the block, we used a filter size of 64, 64, and 128, respectively. The kernel size was 3×3 for all the

TABLE IV

MEAN AND STANDARD DEVIATION OF RMSE FOR HIP, KNEE, AND ANKLE ANGLES OF ALL MODELS FOR ALL WALKING SCENARIOS.

Family	Model	Hip (°)	Knee (°)	Ankle (°)	Mean (°)	Family Mean (°)
Base	Conv2D-Base-Net	6.51 ± 0.99	8.44 ± 1.19	4.91 ± 0.62	6.62 ± 0.93	5.62 ± 0.74
	Bi GRU-Base-Net	5.23 ± 0.66	6.07 ± 0.64	4.09 ± 0.32	5.13 ± 0.54	
	Hybrid Conv1D-GRU-Base-Net	5.25 ± 0.76	6.24 ± 1.17	4.14 ± 0.38	5.21 ± 0.77	
	Hybrid Conv2D-GRU-Base-Net-1	5.11 ± 0.66	5.88 ± 0.83	4.16 ± 0.30	5.05 ± 0.60	
	Hybrid Conv2D-GRU-Base-Net-2	6.25 ± 0.91	7.46 ± 1.13	4.65 ± 0.60	6.12 ± 0.88	
BaseBoost	Conv2D-BaseBoost-Net	6.47 ± 0.91	7.77 ± 1.12	4.75 ± 0.63	6.33 ± 0.88	5.49 ± 0.70
	Bi GRU-BaseBoost-Net	5.19 ± 0.63	5.94 ± 0.72	4.08 ± 0.29	5.07 ± 0.55	
	Hybrid Conv1D-GRU-BaseBoost-Net	5.26 ± 0.68	5.94 ± 0.93	4.09 ± 0.30	5.10 ± 0.64	
	Hybrid Conv2D-GRU-BaseBoost-Net-1	5.15 ± 0.65	5.78 ± 0.75	4.03 ± 0.26	4.99 ± 0.55	
	Hybrid Conv2D-GRU-BaseBoost-Net-2	6.19 ± 0.90	7.21 ± 1.14	4.57 ± 0.61	5.99 ± 0.88	
Bag	Conv2D-Bag-Net	6.51 ± 1.13	8.20 ± 1.15	4.73 ± 0.68	6.48 ± 0.99	5.32 ± 0.75
	Bi GRU-Bag-Net	4.99 ± 0.67	5.57 ± 0.82	3.89 ± 0.30	4.82 ± 0.60	
	Hybrid Conv1D-GRU-Bag-Net	4.95 ± 0.74	5.84 ± 1.05	3.94 ± 0.34	4.91 ± 0.71	
	Hybrid Conv2D-GRU-Bag-Net-1	4.91 ± 0.75	5.48 ± 0.85	3.83 ± 0.33	4.74 ± 0.64	
	Hybrid Conv2D-GRU-Bag-Net-2	5.92 ± 0.85	6.79 ± 1.03	4.30 ± 0.55	5.67 ± 0.81	
BagBoost	Conv2D-BagBoost-Net	6.40 ± 1.01	7.53 ± 1.18	4.54 ± 0.66	6.15 ± 0.95	5.22 ± 0.73
	Bi GRU-BagBoost-Net	4.97 ± 0.72	5.56 ± 0.90	3.87 ± 0.32	4.80 ± 0.65	
	Hybrid Conv1D-GRU-BagBoost-Net	4.97 ± 0.66	5.58 ± 0.87	3.90 ± 0.29	4.82 ± 0.61	
	Hybrid Conv2D-GRU-BagBoost-Net-1	4.93 ± 0.74	5.45 ± 0.81	3.80 ± 0.34	4.73 ± 0.63	
	Hybrid Conv2D-GRU-BagBoost-Net-2	5.89 ± 0.84	6.73 ± 1.03	4.27 ± 0.61	5.63 ± 0.83	
WAE	Base-WAE-Net	**4.89 ± 0.69**	5.60 ± 0.88	3.86 ± 0.30	4.78 ± 0.62	**4.74 ± 0.63**
	BaseBoost-WAE-Net	4.96 ± 0.68	5.51 ± 0.83	3.82 ± 0.31	4.76 ± 0.61	
	Bag-WAE-Net	4.92 ± 0.75	5.48 ± 0.87	3.77 ± 0.32	4.73 ± 0.65	
	BagBoost-WAE-Net	4.93 ± 0.74	5.45 ± 0.85	3.76 ± 0.34	4.72 ± 0.64	
	DeepBBWAE-Net	4.90 ± 0.72	**5.44 ± 0.86**	**3.76 ± 0.32**	**4.70 ± 0.63**	

convolutional layers. For maxpooling2D, we used a pool size of 2×2 for each layer. For the fully-connected block, we used a neuron count of 128 and 64 for two FC layers, respectively. A dropout rate of 0.5 was used for each dropout layer.

Bi-GRU-Net. Cell numbers 128 and 64 were used in each of the two Bi-GRU blocks. A dropout rate of 0.5 was used for each dropout layer.

Hybrid Conv1D-GRU-Net. The Bi-GRU block used the same parameters that were used in Bi-GRU-Net. For the 1D convolutional block, filter sizes of 64, 64, 128, and 128 were used for four convolutional layers with a kernel size of 3. A pool size of 2 was used for maxpooling1D. For a fully connected block, the neuron count was 128 and 64 for the dense layer, and a dropout rate was chosen 0.5 for both of the dropout layers.

Hybrid Conv2D-GRU-Net-1. The parameters for Hybrid Conv2D-GRU-Net-1 were the same as those used in the respective components in base learners 1 and 2.

Hybrid Conv2D-GRU-Net-2. We used the same parameters in Hybrid Conv2D-GRU-Net-1 as was used in base learners 1 and 2.

Hyperparameters of the Base Learners. We used mean squared error as the loss function and an Adam [41] optimizer to optimize the parameters of all the base learners. A batch size of 64 was selected for all the base learners. The number of iterations for each base learner was 250, 70, 200, 200, 70, respectively.

D. Evaluation Methods

For the evaluation, we considered the leave-out-one subject cross-validation method. Excluding a single subject's data from the training set allows for evaluation methods that check for model overfitting and assess the model's overall performance. To measure the accuracy of the predictions, we considered two metrics: RMSE and Pearson correlation coefficient (PCC). RMSE determines the offset between the ground truth and the prediction, where the PCC determines the correlation of ground truth and predicted curves. To show the accuracy of our proposed approach, we take the average of angles of both legs to get single RMSE and correlation values for the hip, knee, and ankle angles on the sagittal plane under each walking test condition.

E. Results

Table IV, V present RMSE and PCC values for all the base learners, intermediate models, and final proposed models for each joint angle under all walking conditions, as well as the overall mean values for each specific model. The models are categorized into five families: Base, BaseBoost, Bag, BagBoost, and WAE. To show the comparative performance of each family, we also reported the family means. We considered the five base models as the baseline for this study, so results from the base models are compared to intermediate and final proposed (DeepBBWAE-Net) models to show the efficacy of the proposed methods.

While the Conv2D-Base-Net model had an RMSE of 6.62° the use of boosting in the Base model, generating the Conv2D-BaseBoost-Net model, resulted in a 4.38% reduction in mean RMSE. Applying bagging techniques on the training data resulted in a 2.11% reduction in mean RMSE compared to the Conv2D-Base-Net model. Using the same boosting methods on the validation data after bagging, as done in the Conv2D-BagBoost-Net model, reduced the mean RMSE by 7.10%. Fig. 6 demonstrates the percent reduction of RMSE and improvement of PCC when bagging, boosting are applied to the base learners, indicating the effectiveness of adding different components to the base learners implemented in our proposed workflow.

After creating BaseBoost, Bag, and BagBoost models from the base learners, we used weighted average ensembles to combine those models. Base-WAE-Net, BaseBoost-WAE-Net, Bag-WAE-Net, and BagBoost-WAE-Net were created using the weighted average ensemble of the Base, BaseBoost, Bag, BagBoost family models, respectively. Fig. 7 demonstrates the effectiveness of the WAE family models over the base learners.

Table IV, V demonstrate the gradual improvements in both families mean RMSE and correlation values as bagging and boosting are added to the models. Overall, the WAE family reduces the mean RMSE by 15.66%, 13.66%, 10.90%, and 9.20% compared to the

TABLE V

MEAN AND STANDARD DEVIATION OF PCC FOR HIP, KNEE, AND ANKLE ANGLE OF ALL MODELS FOR ALL WALKING SCENARIOS.

Family	Model	Hip	Knee	Ankle	Mean	Family Mean
Base	Conv2D-Base-Net	0.931 ± 0.015	0.95 ± 0.014	0.917 ± 0.012	0.933 ± 0.013	0.948 ± 0.009
	Bi GRU-Base-Net	0.951 ± 0.005	0.975 ± 0.004	0.945 ± 0.005	0.957 ± 0.005	
	Hybrid Conv1D-GRU-Base-Net	0.951 ± 0.006	0.974 ± 0.006	0.943 ± 0.006	0.956 ± 0.006	
	Hybrid Conv2D-GRU-Base-Net-1	0.952 ± 0.008	0.975 ± 0.004	0.947 ± 0.003	0.958 ± 0.005	
	Hybrid Conv2D-GRU-Base-Net-2	0.934 ± 0.017	0.957 ± 0.016	0.922 ± 0.013	0.938 ± 0.015	
BaseBoost	Conv2D-BaseBoost-Net	0.932 ± 0.014	0.952 ± 0.013	0.919 ± 0.013	0.934 ± 0.013	0.949 ± 0.009
	Bi GRU-BaseBoost-Net	0.953 ± 0.005	0.975 ± 0.005	0.946 ± 0.005	0.958 ± 0.005	
	Hybrid Conv1D-GRU-BaseBoost-Net	0.952 ± 0.005	0.974 ± 0.006	0.944 ± 0.005	0.957 ± 0.006	
	Hybrid Conv2D-GRU-BaseBoost-Net-1	0.953 ± 0.007	0.976 ± 0.004	0.948 ± 0.003	0.959 ± 0.005	
	Hybrid Conv2D-GRU-BaseBoost-Net-2	0.934 ± 0.016	0.957 ± 0.015	0.924 ± 0.013	0.939 ± 0.015	
Bag	Conv2D-Bag-Net	0.937 ± 0.017	0.954 ± 0.013	0.927 ± 0.013	0.939 ± 0.014	0.956 ± 0.008
	Bi GRU-Bag-Net	0.959 ± 0.005	0.978 ± 0.005	0.954 ± 0.004	0.964 ± 0.004	
	Hybrid Conv1D-GRU-Bag-Net	0.96 ± 0.006	0.978 ± 0.005	0.951 ± 0.004	0.963 ± 0.005	
	Hybrid Conv2D-GRU-Bag-Net-1	0.961 ± 0.005	0.979 ± 0.004	0.956 ± 0.004	0.966 ± 0.004	
	Hybrid Conv2D-GRU-Bag-Net-2	0.944 ± 0.013	0.964 ± 0.012	0.936 ± 0.011	0.948 ± 0.012	
BagBoost	Conv2D-BagBoost-Net	0.938 ± 0.016	0.956 ± 0.012	0.929 ± 0.012	0.941 ± 0.014	0.956 ± 0.008
	Bi GRU-BagBoost-Net	0.959 ± 0.005	0.978 ± 0.004	0.955 ± 0.004	0.964 ± 0.004	
	Hybrid Conv1D-GRU-BagBoost-Net	0.96 ± 0.006	0.978 ± 0.005	0.952 ± 0.004	0.963 ± 0.005	
	Hybrid Conv2D-GRU-BagBoost-Net-1	0.961 ± 0.005	0.979 ± 0.004	0.956 ± 0.004	0.966 ± 0.004	
	Hybrid Conv2D-GRU-BagBoost-Net-2	0.945 ± 0.013	0.964 ± 0.012	0.936 ± 0.011	0.948 ± 0.012	
WAE	Base-WAE-Net	0.959 ± 0.006	0.978 ± 0.004	0.954 ± 0.003	0.964 ± 0.005	**0.965 ± 0.004**
	BaseBoost-WAE-Net	0.959 ± 0.006	0.979 ± 0.004	0.955 ± 0.003	0.964 ± 0.004	
	Bag-WAE-Net	0.961 ± 0.005	0.979 ± 0.004	0.956 ± 0.004	0.966 ± 0.004	
	BagBoost-WAE-Net	0.961 ± 0.005	0.979 ± 0.004	0.957 ± 0.004	0.966 ± 0.004	
	DeepBBWAE-Net	**0.961 ± 0.005**	**0.979 ± 0.004**	**0.957 ± 0.003**	**0.966 ± 0.004**	

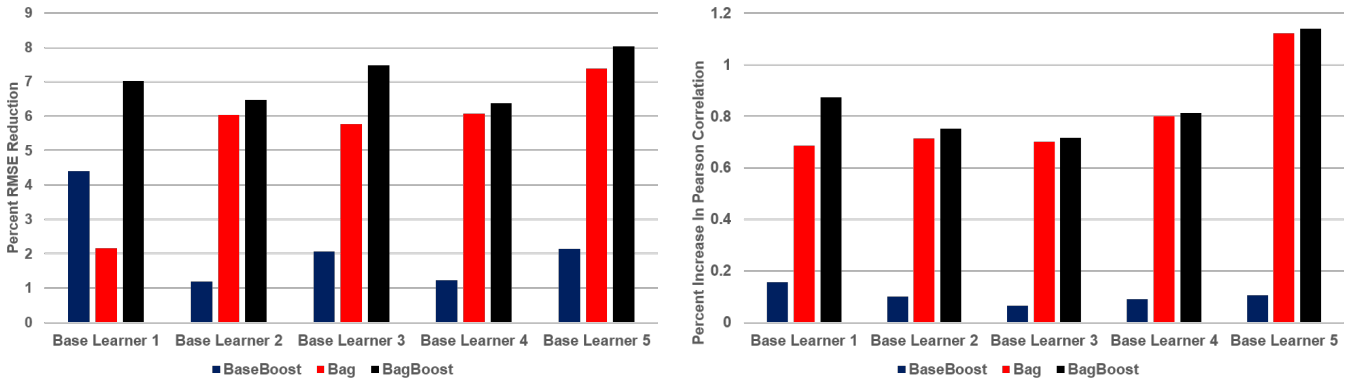

Fig. 6. Percentage RMSE reduction (left figure) and percentage PCC improvement (right figure) over base learners when different techniques are applied (Boost, Bag, BagBoost)

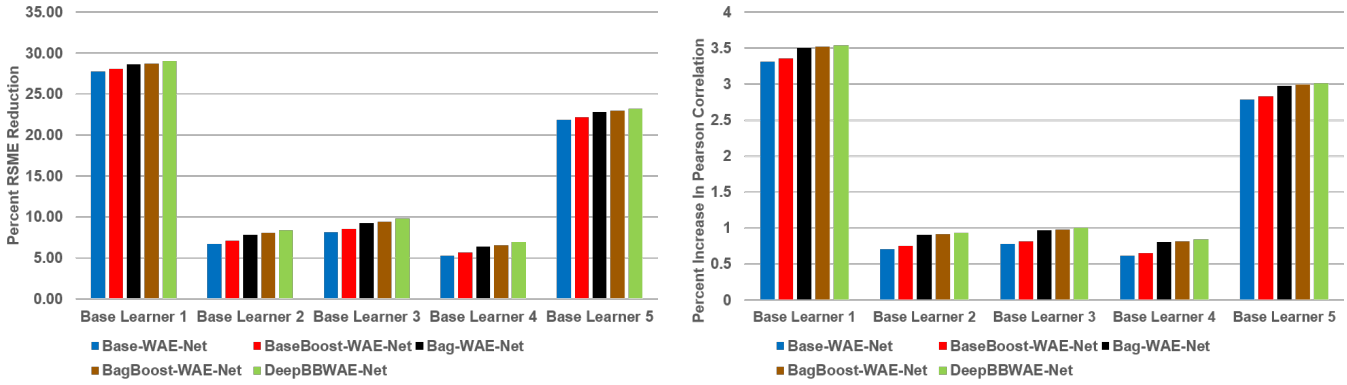

Fig. 7. Percentage RMSE reduction (left figure) and percentage PCC improvement (right figure) over base learners with different WAE family models

Base, BaseBoost, Bag, BagBoost families, respectively while increasing the Pearson correlation coefficients by 1.76%, 1.66%, 0.95%, and 0.90% respectively.

Table VI demonstrates the RMSE and Pearson correlation coefficients of all of the base models averaged, the intermediate models averaged, and of DeepBBWAE-Net for different walking conditions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3165383, IEEE Journal of Biomedical and Health Informatics

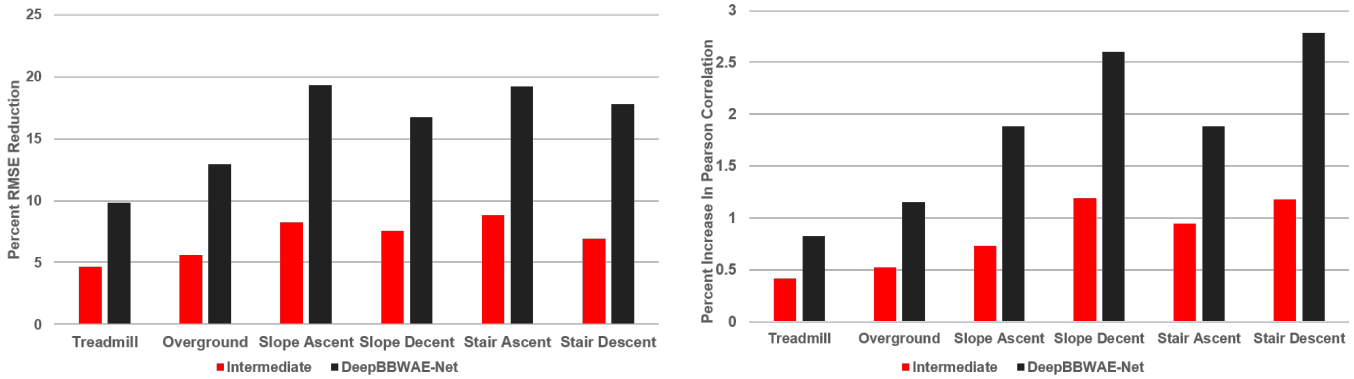

Fig. 8. Percent of RMSE reduction (left figure) and percentage PCC improvement (right figure) of a base learners with Intermediate models and DeepBBWAE-Net

TABLE VI
RMSEs AND PCCs OF BASE MODELS, INTERMEDIATE MODELS, AND DEEPBBWAE-NET FOR DIFFERENT WALKING CONDITIONS

	Model	Treadmill	Overground	Slope Ascent	Slope Descent	Stair Ascent	Stair Descent
RMSE (°)	Base Models	4.29	4.48	5.81	5.68	6.57	6.92
	Intermediate Models	4.09	4.23	5.33	5.25	5.99	6.44
	DeepBBWAE-Net	3.87	3.90	4.69	4.73	5.31	5.69
Correlation	Base Models	0.966	0.957	0.955	0.922	0.955	0.935
	Intermediate Models	0.970	0.962	0.962	0.933	0.964	0.946
	DeepBBWAE-Net	0.974	0.968	0.973	0.946	0.973	0.961

Figure 8 shows the percent improvement of the intermediate models' and DeepBBWAE-Net's predictions over those of the base models.

Figure 9 compares DeepBBWAE-Net kinematic predictions of two gait cycles for each walking condition with the motion capture based kinematics and present their corresponding RMSEs and PCCs to demonstrate the variance between the prediction and ground truth. Table VII shows the RMSEs and PCCs of the DeepBBWAE-Net predictions for each joint for all walking conditions.

V. DISCUSSION

A Novel DeepBBWAE-Net. This paper proposes a novel framework, DeepBBWAE-Net, by implementing bagging, boosting, and weighted average ensemble techniques to estimate joint angles of the lower extremities during gait in different environments using two IMU sensors. The reduced RMSE indicates DeepBBWAE-Net's ability to calculate joint kinematics more accurately than other simpler deep learning models. DeepBBWAE-Net's predictions are stable among the participants with a standard deviation of 0.63° for RMSE. Our results demonstrate that walking conditions with significantly more variability lead to predictions with greater RMSE and lower Pearson correlation coefficients. For example, the joint kinematics of walking is highly dependent on walking speed [42]. The treadmill trials had a controlled constant walking speed throughout the data collection period. On the other hand, when the participants walked back and forth across 5 meter walkways for the overground and sloped walking conditions, there was more variation in walking speed leading to increased joint kinematic variability. Similarly, The variety of walking strategies for stair walking led to predictions with higher RMSEs and lower Pearson correlation coefficients than those of kinematic predictions of other walking conditions. However, DeepBBAWE-Net achieved better joint kinematic predictions than conventional deep learning models under these more variable walking patterns (Fig. 8).

Stair and Slope Kinematics. Many studies have implemented machine learning to estimate kinematics using IMU sensors [14], [19], [20], [23] for overground or treadmill conditions. But, no study has developed machine learning-based estimation of joint kinematics for stair and slope walking. Our final goal is to estimate joint kinematics outside the lab in different walking environments. Training an algorithm with data from stair and slope conditions is a big step toward achieving that goal.

Predictions of the treadmill and overground conditions had lower RMSEs, while the slope and stair condition predictions produced greater error (Table VI). One reason for this is the greater number of consecutive data sets under each controlled walking speed for the treadmill and overground conditions compared to the slope and stair conditions. This is further compounded by the fact that the stair and slope trials have greater kinematic variability. These variances in the kinematics and reduced data set size makes determining the specific window size for input data to the algorithm difficult.

Despite new data management and our new algorithm, the stair condition predictions had greater errors than other walking conditions. The increased error is likely due to the variations in strategies we observed with subjects in the stair walking condition. We found that some participants used their forefoot to climb up the stairs while other participants used their entire foot. Some participants contacted their toe first while descending the stairs, while other participants used their whole foot to land on the stair. This increased inter-subject variability in the stair data set might have made it difficult for the deep learning model to perform good predictions for those conditions.

Another reason for error in slope and stair kinematic predictions was the data sets' size. The slope and stair conditions have less data compared to the those of the treadmill and overground conditions due to the limited number of stairs and the length of the ramp that could fit within the limited capture volume of the motion capture system (Table III). Another factor influencing prediction accuracy of stair and slope conditions was variable participant walking speed. The slope and stair conditions were only performed at a single self-selected speed, while the treadmill and overground conditions had multiple speeds.

Real-time measurement. One of the advantages of our study is the potential for estimation of joint kinematics in real-time as raw IMU

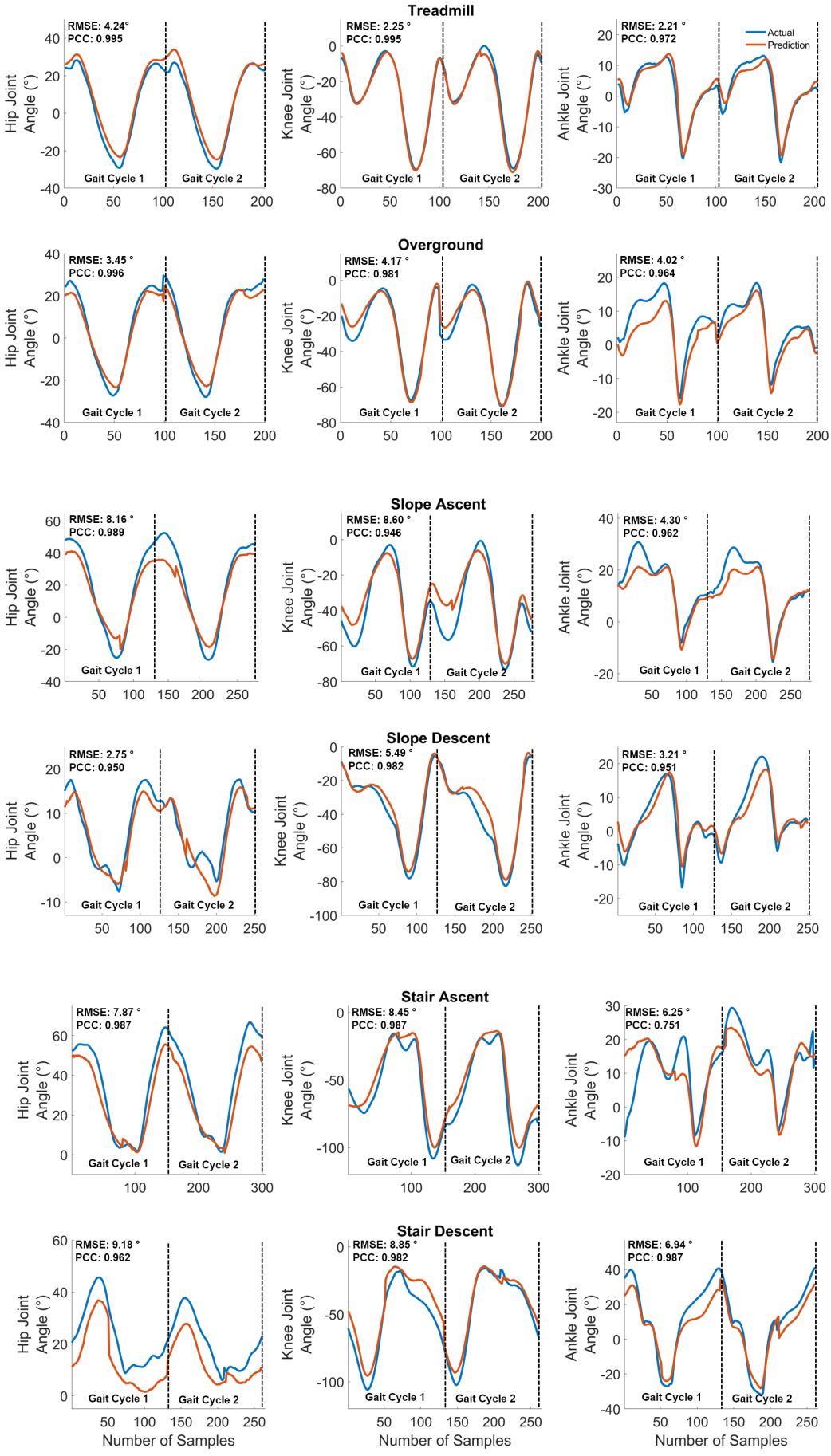

Fig. 9. Sample plots of right hip, knee, and ankle angle for two gait cycles for Subject 2 in all walking scenarios

TABLE VII

MEAN AND STANDARD DEVIATION OF RMSE AND PCC FOR DEEPBBWAE-NET FOR DIFFERENT WALKING CONDITION

	RMSE (°)				PCC			
Walking Condition	**Hip**	**Knee**	**Ankle**	**Mean**	**Hip**	**Knee**	**Ankle**	**Mean**
Treadmill	3.77 ± 0.76	4.62 ± 1.05	3.22 ± 0.75	3.87 ± 0.58	0.984 ± 0.006	0.984 ± 0.009	0.955 ± 0.013	0.974 ± 0.007
Overground	4.32 ± 0.65	4.28 ± 0.69	3.09 ± 0.57	3.90 ± 0.46	0.971 ± 0.010	0.985 ± 0.005	0.949 ± 0.011	0.968 ± 0.004
Slope Ascent	5.65 ± 1.19	4.94 ± 0.85	3.47 ± 0.55	4.69 ± 0.78	0.98 ± 0.004	0.976 ± 0.005	0.962 ± 0.011	0.973 ± 0.020
Slope Descent	4.33 ± 0.59	6.11 ± 1.57	3.74 ± 0.32	4.73 ± 0.65	0.927 ± 0.021	0.972 ± 0.007	0.941 ± 0.016	0.946 ± 0.012
Stair Ascent	6.01 ± 1.51	5.89 ± 1.37	4.01 ± 0.66	5.31 ± 1.05	0.978 ± 0.004	0.986 ± 0.003	0.955 ± 0.015	0.973 ± 0.006
Stair Descent	5.29 ± 1.53	6.78 ± 1.74	5.02 ± 1.36	5.69 ± 1.39	0.929 ± 0.021	0.974 ± 0.016	0.980 ± 0.006	0.961 ± 0.014

TABLE VIII

COMPARISON OF PREVIOUS STUDIES WITH OUR PROPOSED METHOD

Paper	**Mundt et al. [19]**	**Dorschky et al. [20]**	**Gholami et al. [22]**	**Lim et al. [21]**	**Ours**
Number of sensors	5 IMUs	4 IMUs	1 accelerometer	1 IMU	2 IMUs
Placement of sensor	hip, thigh (2), shank (2)	hip, thigh, shank, foot	Foot	Lower back	Feet
Prediction Model	ANN	2D CNN	1D CNN	ANN	DeepBBWAE-Net
Walking Type	walking	walking and running	running	walking	walking
Walking condition	overground	overground	treadmill	treadmill	treadmill, overground
Hip (RMSE)	-	5.08°	5.6°	3.1° (stance phase)	3.77°, 4.32°
Knee (RMSE)	4.62°	4.81°	6.5°	2.2° (stance phase)	4.62°, 4.28°
Ankle (RMSE)	2.42°	4.60°	4.7°	3.4° (stance phase)	3.22°, 3.09°

data is used as input to the model. Previous studies would typically pre-process the collected sensor signal and motion data before input to a machine learning algorithm. For example, gait cycles would be identified and then used to normalized the time data to unify the input data size [19], [21]. Also, input features are usually extracted from the collected data to use as algorithm input [21]. This pre-processing increases computation time, slowing the predictions and reducing the network's viability as a real-time system, preventing the monitoring of kinematics in real-time. Our proposed model uses raw IMU data, removing the need for computationally expensive pre-processing and enables application in real-time kinematic estimation with approximately 6ms inference time for 0.8s input windows.

Comparison with Other Studies. Table VIII compares our results with those of previous studies that use true IMU data and leave out cross validation. These comparisons are only for treadmill and overground conditions, following the limits of what these studies implement. Our model outperforms most of the previous work, but direct comparison of this study's result with others may not be fair as the data set, sensor position, sensor number, and trial conditions are different. The data set used in this study consisted of a combination of treadmill, overground, stair, and slope walking data. While perhaps a more robust representation of human gait, using a combination of different conditions may produce less accurate predictions than a model trained to predict kinematics from the treadmill or overground data only. For this reason, we mainly compare the results of our proposed approach with more conventional models applied to the same data set. Other studies use conventional machine learning or deep learning models (Table VIII) to predict joint kinematics in gait. We have integrated bagging, boosting, and WAE techniques in our framework DeepBBWAE-Net and demonstrated that our proposed approach outperforms conventional methods when trained from the same data set (Table IV,V). This result indicates the advantages of our developed machine learning algorithm over those of other studies to predict joint kinematics in gait more robustly.

Limitation and Future Work. One of the significant challenges addressed in this study is the development of a kinematics estimation approach that requires a reduced number of sensors. This is challenging because the algorithm's input is missing information typically captured with a complete sensor set, potentially affecting the model's accuracy. While adding the input of thigh or shank sensors may improve the performance of kinematics estimation, it makes the system less practical by increasing the potential burden on the subject. One major limitation of our proposed network DeepBBWAE-Net is that it is not trained end-to-end. We train multiple models separately and use a weighted average ensemble to combine their output. This increases the complexity of training the model, so making the framework train end-to-end can be a potential future work. Another limitation of our work is the limited data set. A larger data set would more effectively train the model and result in better kinematic predictions. One potential method for expanding the available data set is to use simulated IMU sensor data [14]. This would be particularly useful for expanding areas of the data set that were less represented, such as stair and slope walking conditions [43]. Using simulation to generate data of hundreds of different artificial subjects may improve the generalization of DeepBBWAE-Net. In the future, we plan to add simulated data to the training and assess the model's performance with the augmented data set. Additionally, we plan on implementing a parameter sweep to identify the optimal number of bootstrap samples (k) for the network.

VI. CONCLUSION

In this paper, we proposed a novel deep learning framework DeepBBWAE-Net to estimate lower extremity kinematics in different walking conditions. Our proposed framework predicts joint kinematics with less error compared to other conventional deep learning models. Our study implements a reduced set of IMU sensors in order to increase the system's viability in commercial systems by increasing user comfort. Our developed algorithm has potential application in the monitoring of the pathologic walking patterns of neurologically impaired individuals patients in their daily living. This would allow clinicians to assess the progression of patient disease and the outcomes of treatment remotely, reducing the frequency of required return visits to an outpatient clinic.

REFERENCES

[1] K. Delbaere, D. L. Sturnieks, G. Crombez, and S. R. Lord, "Concern About Falls Elicits Changes in Gait Parameters in Conditions of Postural Threat in Older People," *The Journals of Gerontology: Series A*, vol. 64A, no. 2, pp. 237–242, 02 2009. [Online]. Available: https://doi.org/10.1093/gerona/gln014

[2] S. A. Bridenbaugh and R. W. Kressig, "Laboratory review: the role of gait analysis in seniors' mobility and fall prevention," *Gerontology*, vol. 57, no. 3, pp. 256–264, 2011.

[3] K. R. Pirker W, "Gait disorders in adults and the elderly : A clinical guide." *Wien Klin Wochenschr.*, vol. 129, no. 3-4, pp. 81–95, 2017.

[4] F. A. Fatoye, M. L. van der Linden, P. J. Rowe, S. Palmer, and F. Macmillan, "Gait kinematics and passive knee joint range of motion in children with hypermobility syndrome," *Gait and Posture*, vol. 33, pp. 447–451, 2011.

[5] K.-S. Shih, S.-Y. Jhou, W.-C. Hsu, C.-C. Hsu, J.-W. Chen, J.-C. Yeh, and Y.-C. Hung, "A biomechanical investigation of athletic footwear traction performance: Integration of gait analysis with computational simulation," *Applied Sciences*, vol. 10, no. 5, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/5/1672

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

[7] N. Hegde, T. Zhang, G. Uswatte, E. Taub, J. Barman, S. McKay, A. Taylor, D. M. Morris, A. Griffin, and E. S. Sazonov, "The pediatric smartshoe: Wearable sensor system for ambulatory monitoring of physical activity and gait," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 477–486, 2018.

[8] G. Marta, F. Simona, C. Andrea, B. Dario, S. Stefano, V. Federico, B. Marco, B. Francesco, M. Stefano, and P. Alessandra, "Wearable biofeedback suit to promote and monitor aquatic exercises: A feasibility study," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1219–1231, 2020.

[9] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "Opensim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[10] B. R., "Gait analysis methods in rehabilitation," *Journal of neuroengineering and rehabilitation*, vol. 3:4, 2006.

[11] I. Weygers, M. Kok, M. Konings, H. Hallez, H. De Vroey, and K. Claeys, "Inertial sensor-based lower limb joint kinematics: A methodological systematic review," *Sensors*, vol. 20, no. 3, 2020.

[12] E. Palermo, S. Rossi, F. Marini, F. Patanè, and P. Cappa, "Experimental evaluation of accuracy and repeatability of a novel body-to-sensor calibration procedure for inertial sensor-based gait analysis," *Measurement*, vol. 52, pp. 145–155, 2014.

[13] T. Zimmermann, B. Taetz, and G. Bleser, "Imu-to-segment assignment and orientation alignment for the lower body using deep learning," *Sensors*, vol. 18, no. 1, 2018.

[14] M. Mundt, A. Koeppe, F. Bamer, S. David, and B. Markert, "Artificial neural networks in motion analysis—applications of unsupervised and heuristic feature selection techniques," *Sensors*, vol. 20, no. 16, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/16/4581

[15] G. Mayer-Kress, Y.-T. Liu, and K. M. Newell, "Complex systems and human movement," *Complexity*, vol. 12, no. 2, pp. 40–51, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cplx.20151

[16] J. C. Alcaraz, S. Moghaddamnia, M. Fuhrwerk, and J. Peissig, "Efficiency of the memory polynomial model in realizing digital twins for gait assessment," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[17] A. Baghdadi, L. A. Cavuoto, and J. L. Crassidis, "Hip and trunk kinematics estimation in gait through kalman filter using imu data at the ankle," *IEEE Sensors Journal*, vol. 18, no. 10, pp. 4253–4260, 2018.

[18] A. Salarian, P. R. Burkhard, F. J. G. Vingerhoets, B. M. Jolles, and K. Aminian, "A novel approach to reducing number of sensing units for wearable gait analysis systems," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 1, pp. 72–77, 2013.

[19] M. Mundt, A. Koeppe, S. David, T. Witter, F. Bamer, W. Potthast, and B. Markert, "Estimation of gait mechanics based on simulated and measured imu data using an artificial neural network," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 41, 2020.

[20] E. Dorschky, M. Nitschke, C. F. Martindale, A. J. van den Bogert, A. D. Koelewijn, and B. M. Eskofier, "Cnn-based estimation of sagittal plane walking and running biomechanics from measured and simulated inertial sensor data," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 604, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fbioe.2020.00604

[21] H. Lim, B. Kim, and S. Park, "Prediction of lower limb kinetics and kinematics during walking by a single imu on the lower back using machine learning," *Sensors*, vol. 20, no. 1, 2020.

[22] M. Gholami, C. Napier, and C. Menon, "Estimating lower extremity running gait kinematics with a single accelerometer: A deep learning approach," *Sensors*, vol. 20, no. 10, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/10/2939

[23] M. Mundt, W. Thomsen, T. Witter, A. Koeppe, S. David, F. Bamer, W. Potthast, and B. Markert, "Prediction of lower limb joint angles and moments during gait using artificial neural networks," *Medical amp; biological engineering amp; computing*, vol. 58, no. 1, p. 211—225, January 2020. [Online]. Available: https://doi.org/10.1007/s11517-019-02061-3

[24] V. Hernandez, D. Dadkhah, V. Babakeshizadeh, and D. Kulić, "Lower body kinematics estimation from wearable sensors for walking and running: A deep learning approach," *Gait Posture*, vol. 83, pp. 185–193, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0966636220306068

[25] E. Rapp, S. Shin, W. Thomsen, R. Ferber, and E. Halilaj, "Estimation of kinematics from inertial measurement units using a combined deep learning and optimization framework," *Journal of Biomechanics*, vol. 116, p. 110229, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021929021000099

[26] J. C. Alcaraz, S. Moghaddamnia, and J. Peissig, "Efficiency of deep neural networks for joint angle modeling in digital gait assessment," *EURASIP J. Adv. Signal Process*, vol. 10 (2021), 2021.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[28] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[29] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, p. 123–140, Aug. 1996. [Online]. Available: https://doi.org/10.1023/A:1018054314350

[30] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *The Annals of Statistics*, vol. 29, no. 5, pp. 1189 – 1232, 2001.

[31] K. S. D Kraft, "SLSQP—a nonlinear programming method with quadratic programming subproblems," DLR, Oberpfaffenhofen, 1989.

[32] F. Li, K. Shirahama, M. A. Nisar, X. Huang, and M. Grzegorzek, "Deep transfer learning for time series data based on sensor modality classification," *Sensors*, vol. 20, no. 15, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4271

[33] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[35] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks – ICANN 2010*, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101.

[36] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[37] K. Karamanidis and A. Arampatzis, "Evidence of mechanical load redistribution at the knee joint in the elderly when ascending stairs and ramps," *Annals of biomedical engineering*, vol. 37, no. 3, p. 467—476, March 2009.

[38] I. Komnik, M. Peters, J. Funken, S. David, S. Weiss, and W. Potthast, "Non-sagittal knee joint kinematics and kinetics during gait on level and sloped grounds with unicompartmental and total knee arthroplasty patients," *PLOS ONE*, vol. 11, no. 12, pp. 1–18, 12 2016. [Online]. Available: https://doi.org/10.1371/journal.pone.0168566

[39] M. Q. Liu, F. C. Anderson, M. H. Schwartz, and S. L. Delp, "Muscle contributions to support and progression over a range of walking speeds," *Journal of Biomechanics*, vol. 41, no. 15, pp. 3243–3252, 2008.

[40] R. H. K. . W. M. E. Kadaba, M. P., "Measurement of lower extremity kinematics during level walking," *Journal of Orthopaedic Research : Official Publication of the Orthopaedic Research Society*, vol. 8, no. 3, pp. 383–392, 1990.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[42] M. H. Schwartz, A. Rozumalski, and J. P. Trost, "The effect of walking speed on the gait of typically developing children," *Journal of Biomechanics*, vol. 41, no. 8, pp. 1639–1650, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021929008001450

[43] T. Lencioni, I. Carpinella, M. Rabuffetti, A. Marzegan, and M. Ferrarin, "Human kinematic, kinetic and emg data during different walking and stair ascending and descending tasks," *Scientific data*, vol. 6, no. 1, Dec. 2019.