IMU-based Deep Neural Networks for Locomotor Intention Prediction

Lu, Huaitian; Schomaker, Lambert; Carloni, Raffaella

Published in:
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

DOI:
10.1109/IROS45743.2020.9341649

Publication date:
2020

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

IMU-based Deep Neural Networks for Locomotor Intention Prediction

Huaitian Lu, Lambert R.B. Schomaker, and Raffaella Carloni

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-7281-6212-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/IROS45743.2020.9341649

Abstract— **This paper focuses on the design and comparison of different deep neural networks for the real-time prediction of locomotor intentions by using data from inertial measurement units. The deep neural network architectures are convolutional neural networks, recurrent neural networks, and convolutional recurrent neural networks. The input to the architectures are features in the time domain, which have been derived either from one inertial measurement unit placed on the upper right leg of ten healthy subjects, or two inertial measurement units placed on both the upper and lower right leg of ten healthy subjects. The study shows that a WaveNet, i.e., a full convolutional neural network, achieves a peak F1-score of 87.17% in the case of one IMU, and a peak of 97.88% in the case of two IMUs, with a 5-fold cross-validation.**

I. Introduction

The accurate prediction of locomotor intention is a fundamental step towards the achievement of the intuitive control of lower limbs prostheses. To avoid discomfort in the use of the prosthetic leg, to reduce the user's cognitive load, and to guarantee safety, the locomotor intention should be predicted and converted to the correct control mode within 300 ms [1].

Inertial measurement units (IMUs) have been used for the prediction of locomotion modes and as the control input for lower limb powered prostheses [2]. To translate the information contained in the IMUs signals into a locomotion mode, a number of data analysis and machine learning techniques have been proposed that are able to recognize patterns in the IMUs signals in real-time. Specifically, the existing IMU pattern recognition approaches can be divided into two categories, i.e., methods based on feature engineering [3] and methods based on feature learning [4], with either hand-crafted input data or raw input data.

For feature engineering, methods have been studied for locomotion mode recognition and locomotion intent prediction. Figueiredo *et al.* used hand-crafted features from several IMUs, and compared different supervised machine learning classifiers, i.e., discriminant analysis, k-nearest neighbors algorithm, random forest, support-vector machine, and multilayer perceptron [5]. Other research has focussed on the use of hand-crafted features from IMUs and mechanical sensors' data [6], IMUs and force sensors' data [7], [8], IMU and pressure sensors' data [9], to name few.

IMU features learning, by means of deep learning methods, has also been recently used for locomotion mode recognition and locomotion intent prediction. For example, deep belief networks have been used in combination with the spectrogram of one accelerometer sensor [10], Convolutional Neural Networks (CNNs) with one IMU on the foot [11], [12], CNN with several IMUs placed at different locations on the lower-limbs and torso [13], [14], [15], Recurrent Neural Network (RNN) with one IMU on the lower back [16].

This paper focuses on the real-time prediction of locomotor intentions by means of deep neural networks by using data from IMUs. Nine different artificial neural network architectures, based on CNNs, RNNs, and Convolutional Recurrent Neural Networks (CRNNs), are designed and compared. The inputs to the architectures are features in the time domain, which have been obtained from either one IMU (placed on the upper right leg of ten healthy subjects) or two IMUs (placed on both the upper and lower right leg of ten healthy subjects). Specifically, the inputs are IMU raw data (i.e., angular accelerations and angular velocities, obtained from 3-axis accelerometers and 3-axis gyroscopes) and quaternions (i.e., the attitude of the upper and/or lower leg, estimated from the IMU raw data). The task concerns the prediction of seven locomotion actions, i.e., sitting, standing, ground-level walking, ramp ascent and descent, stair ascent and descent. The study shows that a WaveNet, i.e., a full CNN, achieves an average F1-score of 83.0% (with standard deviation of 0.052) in the case of one IMU, and an average F1-score of 95.58% (with standard deviation of 0.05) in the case of two IMUs, with a 5-fold cross-validation. Moreover, the WaveNet achieves a peak F1-score of 87.17% in the case of one IMU, and a peak of 97.88% in the case of two IMUs, with a 5-fold cross-validation.

The remainder of the paper is organized as follows. In Section II, the materials and methods for the prediction of the locomotor intention are described. In Section III, the results of the study are presented and discussed. Finally, concluding remarks are drawn in Section IV.

II. Materials and Methods

This Section presents the design of nine different deep neural networks architectures for the real-time prediction of seven locomotor actions, based on the raw data collected either from one IMU sensor (placed on the upper right leg of ten healthy subjects) or from two IMUs (placed on the upper and lower right leg of ten healthy subjects).

A. Data-set

The data used in this study is the Encyclopedia of Able-bodied Bilateral Lower Limb Locomotor Signals (ENABL3S) public data-set [17]. The data have been collected on ten healthy subjects, i.e., seven males and three females,

This work was funded by the European Commission's Horizon 2020 Programme as part of the project MyLeg under grant no. 780871.

The authors are with the Faculty of Science and Engineering, Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, The Netherlands. atianhlu@gmail.com, {l.r.b.schomaker, r.carloni}@rug.nl

with an average age of 25.5 ± 2 years, height of 174 ± 12 cm, and weight of 70 ± 14 kg. From the ENABL3S data-set, this study only uses the data from the two IMUs on the upper and lower right leg. The IMU raw data are sampled with a sampling frequency of 500 Hz.

The locomotion actions that need to be predicted are S: Sitting, St: Standing, LW: Ground Level Walking, RA: Ramp Ascent, RD: Ramp Descent, SA: Stair Ascent, SD: Stair Descent. During the recording, each subject performed the same locomotion actions in the same order, i.e., the odd circuit is: S → St → LW → SA → LW → RD → LW → St → S; the even circuit is: S → St → LW → RA → LW → SD → LW → St → S. The odd circuit includes stair ascent and ramp descent, the even circuit includes ramp ascent and stair descent. The stairs consist of four steps and the ramps have slopes of 10°. By using a key-fob, the data labeling is done on the true locomotor intention.

B. Input

1) Features: The inputs to the deep neural networks are extracted from the IMUs. Specifically, two different scenarios are compared: (i) one IMU on the upper right leg; (ii) two IMUs, one on the upper right leg and one on the lower right leg. The features used in this study are the raw IMU data (i.e., rotational accelerations and rotational velocities of the upper and/or lower leg, respectively obtained from 3-axis accelerometers and a 3-axis gyroscopes) and the quaternions (i.e., the attitude of the upper and/or lower right leg). To estimate the quaternions, the filter proposed in [18], with the implementation presented in [19], has been used. The choice of these input data is inspired by [20].

2) Sample Generation: The original IMU data are sampled with a frequency of 500 Hz, i.e., each data frame is available every 2 ms. For each healthy subject, 10 adjacent data frames are sequentially concatenated in one sample using a sliding window (with stride equal to 3) method.

3) Scaling: The data have been standardized within each sample, by centering to the mean and by scaling element-wise to the unit variance.

4) Data Partitioning: The data-set has been divided as follows: 80% for training and 20% for testing. Within the training set, 10% was also used for validation. This was done to prevent overfitting of the neural networks on the data.

5) Categorize: The locomotion actions have been categorized, i.e., they have been encoded into a one-hot matrix.

C. Output

The output of the neural networks has dimension equal to seven, i.e., the number of locomotor actions to be predicted (sitting, standing, ground-level walking, ramp ascent and descent, stair ascent and descent).

D. Deep Neural Networks Architectures

Nine deep neural network architectures are designed and compared in this study, and are further described in the following subsections. Specifically, the different architectures are based on CNNs, RNNs and CRNNs.

1) CNNs: Three different CNN architectures (i.e., CNN1D, CNN2D, and WaveNet) have been designed.

Figure 1 shows the CNN1D and the CNN2D, which both consist of six hidden layers, i.e., two convolution layers, two max-pooling layers, and two dense layers. Figure 1 (left) shows the CNN1D architecture. The input is an array of 10 rows (i.e., the frames) and num_features rows. The direction of the convolutional operation is frame-wise, i.e., the convolutional kernel (filter) moves from up to down. The first layer is a convolutional 1D layer, consists of 32 filters, and the length of each kernel is 3. To overcome the numerical problems related to the non-linear threshold functions, a rectified linear unit, i.e., a linear activation function, is used [21]. The next layer is a max-pooling layer, with pooling length of 2. Then, a convolutional 1D layer with 64 filters and kernel length of 3 follows, and the same max-pooling layer follows again. Then, a dense layer with 50 units follows, with 0.25 dropout parameters. Finally, the output layer is a dense layer with 7 units (i.e., num_class) and with a softmax activation function. Figure 1 (right) shows the CNN2D architecture. It is similar to the CNN1D architecture. The most significant difference is that the convolution kernels of the CNN2D slide both row-wise and column-wise. The size of the CNN2D kernel is 3×3, and the size of the max-pooling is 2×2.

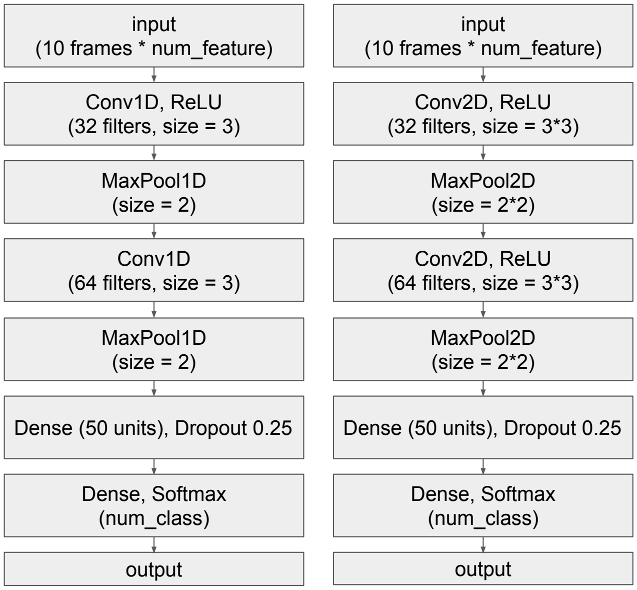

Fig. 1: CNN1D (left) and CNN2D (right) architectures. Both CNN architectures consist of six hidden layers, including two convolution layers, two max-pooling layers, and two dense layers.

Figure 2 shows the WaveNet, i.e., a full CNN, which consists of three convolutional layers. The input is processed by a causal convolutional layer (with 64 filters and filter size 3), then the current output goes through two ways. In the first way, the output goes through a dilated convolutional layer (with 64 filters and filter size 3), to a dot multiplication of the two tanh and sigmoid, and then to the current output layer. In the second way, the output skips the dilated layer and connects to the current output layer directly, sums up

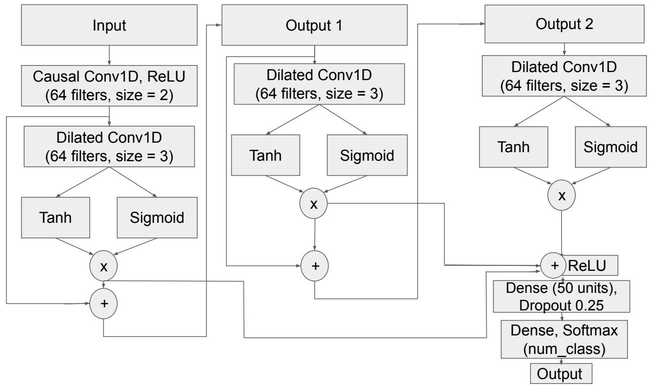

Fig. 2: The WaveNet architecture consists of three convolutional layers.

with the skipped element, and goes to the second layer. A rectified linear unit and a softmax function calculate the final output. The WaveNet architecture proposed in this study is inspired by [22], with the main difference that the WaveNet is used for locomotor intention prediction instead of being a deep generative model of raw audio waveforms.

2) RNNs: Two different RNN architectures have been designed as shown in Figure 3, and they both consists of four hidden layers i.e., two recurrent layers, which can be either long short-term memory (LSTMs) [23] or gated recurrent units (GRUs) [24], and two dense layers.

As shown in Figure 3, the input is a sequence of 10 frames. The first layer consists of 30 LSTMs (or GRUs) networks, and the next layer is identical. Then, a dense layer with 50 units follows, with 0.25 dropout parameters. Finally, the output layer is a dense layer with 7 units (i.e., num_class) and with a softmax activation function. The RNN architectures proposed in this study are inspired by [16], with the main difference that the RNN is used for locomotor intention prediction instead of gait analysis. Moreover, we investigate both LSTM and GRU networks.

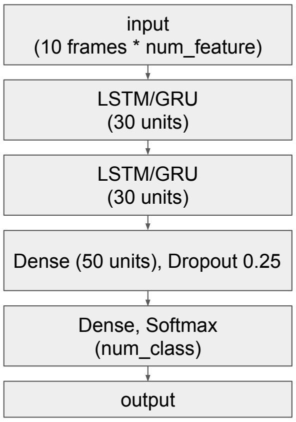

Fig. 3: RNN (either LSTM or GRU) architectures. Both RNN architectures consist of four hidden layers, including two recurrent layers (LSTMs or GRUs) and two dense layers.

3) CRNNs: Four different CRNN architectures have been designed as shown in Figure 3, and they both consists of eight hidden layers, i.e., two convolution layers (either CNN1D or CNN2D), two recurrent layers (either LSTM or GRUs), two max-pooling layers, and two dense layers.

Figure 4 (left) shows the CNN1DLSTM (or CNN1DGRU) architectures, in which the first four layers are the same as in the CNN1D. Then there are two LSTMs (or GRUs), each one with 30 units. A dense layer with 50 units follows, with 0.25 dropout parameters. The output layer is a dense layer with 7 units (i.e., num_class) with a softmax activation function.

Figure 4 (right) shows the CNN2DLSTM (or CNN2DGRU) architectures. The main difference is that, since the interface of the CNN2D and the LSTMs (or GRUs) are not directly compatible, the output of the CNN2D needs to be wrapped together with the time-step to be fed to the LSTM (or GRU). These CRNN architectures are inspired by [25], with the main difference that the RCNN is used for locomotor intention prediction instead of hand position regression. Moreover, we investigate both LSTM and GRU networks.

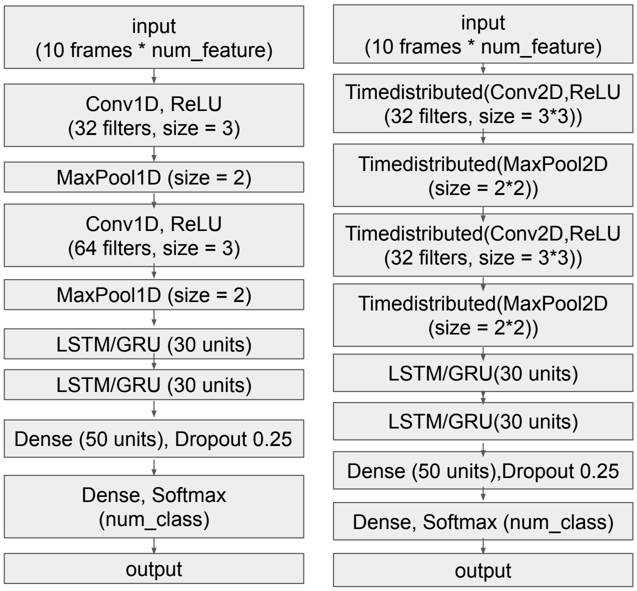
input
(10 frames * num_feature)
Conv1D, ReLU
(32 filters, size = 3)
MaxPool1D (size = 2)
Conv1D, ReLU
(64 filters, size = 3)
MaxPool1D (size = 2)
LSTM/GRU (30 units)
LSTM/GRU (30 units)
Dense (50 units), Dropout 0.25
Dense, Softmax
(num_class)
output
input
(10 frames * num_feature)
Timedistributed(Conv2D,ReLU
(32 filters, size = 3*3))
Timedistributed(MaxPool2D
(size = 2*2))
Timedistributed(Conv2D,ReLU
(32 filters, size = 3*3))
Timedistributed(MaxPool2D
(size = 2*2))
LSTM/GRU(30 units)
LSTM/GRU(30 units)
Dense (50 units),Dropout 0.25
Dense, Softmax (num_class)
output

Fig. 4: CRNN architectures: CNN1DLSTM (or CNN1DGRU) on the left and the CNN2DLSTM (or CNN2DGRU) on the right. The four CRNN architectures consist of eight hidden layers, including two convolution layers, two recurrent layers, two max-pooling layers, and two dense layers.

E. Deep Neural Networks Training

This Section describes how the deep neural network architectures have been trained. The training was done on the cloud computational platform provided by Kaggle (`www.kaggle.com`), with a single K80 GPU and 13 GB RAM.

1) Experiment Procedure: Since nine neural networks are under investigation, in order to simplify the experiment, a random subject has been selected in the ENABL3S data-set (i.e., subject AB156) to perform the first comparison. The three architectures, which performed the best on this subject, have been trained and tested again on the other nine subjects for the final comparison and evaluation.

2) Epoch: The complete data-set is presented 150 times (i.e., epochs) to the neural networks during training to prevent under-fitting and/or ineffective use of training data.

3) Batch Size: The parameters in the neural networks' neurons are updated after a batch of samples is processed. The batch size is chosen to be 128, and the samples of one batch are usually shuffled from the training set. Using a batch approach is equivalent to artificially introducing sampling noise on the gradient, so it is more difficult to fall into a local minimum.

4) Shuffling: The input data have been shuffled when feeding the neural networks. The goal of shuffling is to increase the generalization ability of an architecture. In the training step, the data is usually fed to the architecture batch by batch. The distribution of the samples within a batch could be a problem. If the input data is not shuffled, the training will make the model linger between two over-fittings, and it would not perform well. After shuffling, the distribution of the samples in a batch is closer to the actual distribution of the data, which makes the trained architecture more adaptive on the data. Consequently, it also increases the speed of the convergence of the neural networks.

5) Loss Function: The categorical cross-entropy has been used as the loss function to train the neural networks.

6) Optimizer: The optimizer is the advanced algorithm to optimize the gradient descent and, thus, to speed up the convergence of a neural network and to save the computational power. In this study, the Adaptive Moment Estimation (Adam) has been used as the optimizer [26]. Adam computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients.

7) Learning Rate: The learning rate is initialized to the trade-off value 0.001. A higher learning rate means faster learning, but if it is too high, the architecture can hardly converge. If it is too low, the loss function can fall into a local minimum or the learning time becomes too long.

8) Class Weighting: The amounts of different classes, i.e., the locomotion actions, is imbalanced in the data-set. Therefore, in the training phase, the classes are weighted differently in the loss function, i.e., according to their ratio. This way the neural networks can pay more attention to the under-represented classes.

9) Early Stopping: The number of training epochs is usually set high to ensure the data is fully utilized, and the architecture is not under-fitting. However, continuing the training after the model converges could cause an overfitting. Thus, the early stopping approach has been used, by monitoring the accuracy on the validation set of each epoch. In these experiments, if after 10 epochs the accuracy has not increased, the training stops, and the architecture would be the final. The number of epochs is empirically set slightly larger than the number of the jitters observed in early epochs.

F. Evaluation

The F1-score is the metric used to assess the performance of the neural networks. The F1-score is calculated as:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

with

$$\text{precision} = \frac{tp}{tp + fp}, \qquad \text{recall} = \frac{tp}{tp + fn}$$

where tp is the number of true positive results in the locomotor intention prediction, fp is the number of false positive, and fn is the number of false negative.

The k-fold cross-validation is used to assess the effectiveness of the neural networks. In this work, k is set to 5. This means that the data-set is divided equally into 5 subsets, then the neural network uses four of them as the training set and one as the testing set, so that the process of training and testing is repeated 5 times in total. The final result is the F1-score averaged from the sub-results. This way, every data sample has contributed as training sample, avoiding the waste of data and increasing the reliability of the evaluation since each testing is differently set.

III. Results and Discussion

In this Section, the proposed neural networks are compared according to the F1-score metric. The results are reported separately for the cases of features from only one IMU (on the upper right leg) and two IMUs (one on the upper right leg and one on the lower right leg).

A. One IMU (on the Upper Leg)

Figure 5 shows the F1-scores (mean and standard deviation SD), with a 5-fold cross-validation, of all the designed deep neural networks on the subject AB156, when only features from one IMU on the upper right leg are used. It can be noted that the WaveNet outperforms the other architectures. The CNN2DLSTM and CNN2DGRU architectures also perform well. Thus, the three best architectures, i.e., WaveNet, CNN2DLSTM, and CNN2DGRU, have been selected to be tested on the other nine subjects.

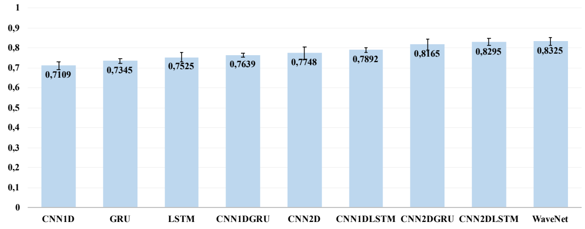

Fig. 5: F1-score (mean and SD), with a 5-fold cross-validation, of all the neural networks on the subject AB156. Only features from one IMU on the upper right leg are used.

Figure 6 shows the F1-scores, with a 5-fold cross-validation, of the three outperforming deep neural networks (i.e., CNN2DGRU, CNN2DLSTM, and WaveNet), on all the ten subjects, when only features from one IMU on the upper right leg are used. The statistic comparison is made using the results of the subjects excluding the subject AB156. The best performing network is the WaveNet, with an average F1-score of 83.0% (with SD of 0.052), which outperforms both the CNN2DLSTM (average F1-score of 81.55%, with SD of 0.064) and the CNN2DGRU (average F1-score of 79.86%, with SD of 0.068). A paired t-test shows that the

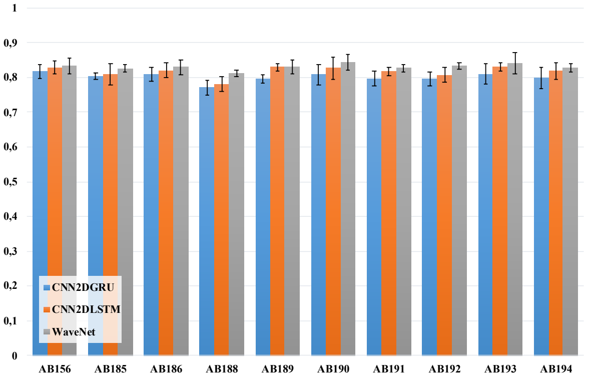

Fig. 6: F1-score (mean and SD), with a 5-fold cross-validation, for the outperforming deep neural networks (i.e., CNN2DGRU, CNN2DLSTM, and WaveNet) per subject. The inputs to the neural networks are data from one IMU on the upper right leg.

WaveNet has a significant difference with respect to the CNN2DLSTM ($p = 0.003 < 0.05$). The CNN2DLSTM has a significant difference with respect to the CNN2DGRU ($p = 0.029 < 0.05$).

B. Two IMUs (on the Upper and Lower Leg)

Figure 7 shows the F1-scores, with a 5-fold cross-validation, of the three deep neural networks (i.e., CNN2DGRU, CNN2DLSTM, and WaveNet), on all the ten subjects, when features from two IMU (one on the upper right leg and one on the lower right leg) are used. Compared to the results with only one IMU, the F1-score increases by 0.12 in average. The statistic comparison is made using the results of the subjects excluding the subject AB156. The best performing network is the WaveNet, with an average F1-score of 95.58% (with SD of 0.05), which outperforms both the CNN2DLSTM (average F1-score of 92.53%, with SD of 0.059) and the CNN2DGRU (average F1-score of 92.0%, with SD of 0.064). A paired t-test shows that the WaveNet has a significant difference with respect to the CNN2DLSTM ($p = 0.006 < 0.05$). The CNN2DLSTM does not have a significant difference with respect to the CNN2DGRU ($p = 0.697 > 0.05$).

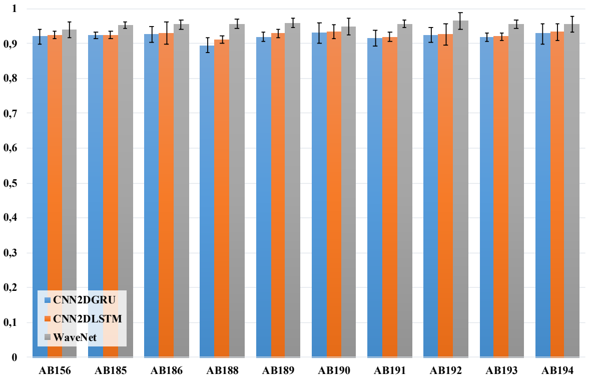

Fig. 7: F1-score (mean and SD) with a 5-fold cross-validation, for the three neural network (i.e., CNN2DGRU, CNN2DLSTM, and WaveNet) per subject. The inputs to the neural networks are data from one IMU on the upper right leg and one IMU on the lower right leg.

C. Running Time

Table I shows the running time (in ms) of the three outperforming deep neural networks when individually classifying 10.000 samples. The samples are randomly chosen over the ten healthy subjects. It can be noted that, when features from two IMUs are used, the classification performance improve significantly, but also the processing time increases.

TABLE I: Average running time (in ms) of three outperforming models when individually classifying 10.000 samples.

IMUs	CNN2DGRU	CNN2DLSTM	WaveNet
Upper leg	7.03 ±0.14	7.61 ±0.15	4.75 ±0.11
Upper & lower leg	7.96 ±0.12	8.84 ±0.13	6.11 ±0.09

D. Results Summary

Table II summarizes the results of the two experimental scenarios (one or two IMUs), which have been analyzed in this study. The WaveNet outperforms all the other deep neural networks in the locomotor intention prediction. Moreover, the running time of the WaveNet is shorter of some milliseconds when compared to the other architectures. The main drawback of the WaveNets is that it took about ten hours to train. However, the training will occur only at the beginning of the use of this deep neural network architecture.

TABLE II: Summary of the results of two experiments, i.e., one IMU (on the upper right leg) and two IMUs (one of the upper right leg and one on the lower right leg).

Locomotion intention prediction	
Features	Raw IMU data + Quaternions
N. of features (per sample)	– 10 (using one IMU): 6 features of raw data and 4 features of quaternions estimated from raw data – 20 (using two IMUs): each IMU uses 6 features of raw data and 4 features of quaternions estimated from raw data
N. of frames (per sample)	10
N. of samples (per subject)	About 0.5 million
N. of locomotion actions	7
Data-set partitioning	5-fold (80% training, 20% testing)
F1-score (mean and SD) for nine subjects (excluding AB156), with one IMU	CNN2DGRU: 79.86%, SD = 0.068 CNN2DLSTM: 81.55%, SD = 0.064 WaveNet: 83.00%, SD = 0.052
F1-score (mean and SD) for nine subjects (excluding AB156), with two IMUs	CNN2DGRU: 92.0%, SD = 0.064 CNN2DLSTM: 92.53%, SD = 0.059 WaveNet: 95.58%, SD = 0.05

Table III shows the confusion matrix of one random result in the experimental scenario with features from one IMU and two IMUs on the subject AB156. It can be noticed that two IMUs can make the locomotor intent recognition more stable among different locomotion actions. The neural networks can generally recognize a sample within a short interval, e.g., WaveNet only used about 4.75 ms to predict an input frame.

E. Discussion

With a peak F1-score of 97.88% for subject AB194, the designed WaveNet, with inputs from two IMUs (one on the

TABLE III: The confusion matrices of one random result, when using WaveNet on the features from one IMU and two IMUs on the subject AB156.

1 IMU	S	LW	RA	RD	SA	SD	St
S	**0.97**	0	0	0	0	0	0.02
LW	0	**0.89**	0.02	0.05	0	0.01	0.02
RA	0	0.09	**0.88**	0	0.01	0	0
RD	0	0.14	0	**0.83**	0	0.01	0.01
SA	0	0.05	0.09	0.02	**0.81**	0	0.02
SD	0	0.07	0	0.0 8	0	**0.83**	0
St	0.05	0.12	0	0.02	0	0	**0.80**

2 IMUs	S	LW	RA	RD	SA	SD	St
S	**0.98**	0	0	0	0	0	0.02
LW	0	**0.96**	0.01	0.02	0	0	0
RA	0	0.02	**0.98**	0	0.01	0	0
RD	0	0.02	0	**0.98**	0	0.01	0.01
SA	0	0.02	0	0	**0.98**	0	0.00
SD	0	0.01	0	0.0 8	0.01	**0.98**	0
St	0.03	0.02	0	0.00	0	0	**0.95**

upper leg and one on the lower limb), has a performance comparable to the machine learning classifiers in [5] and to the CNN in [14], with the major difference that our study uses two IMUs instead of several IMUs. Moreover, the designed WaveNet outperforms: (i) CNNs that use one IMU [11], [12], where 91.97% and 96.7% have been found, respectively; (ii) RNNs that use one IMU [16], where 96.3% has been found; (iii) CNNs that use several IMUs [13], [15], where 97.06%, and 94.15% have been found, respectively.

IV. Conclusion

This paper presented a comparison of different deep neural networks architectures for the real-time locomotor intention prediction. The inputs to the architectures are features in the time-domain from IMU data, i.e., raw data and quaternions. Two scenarios have been compared based on either input data from one IMU (on the upper right leg) or from two IMUs (one on the upper right leg and one on the lower right leg). The architectures have to predict seven locomotion actions, i.e., sitting, standing, ground-level walking, ramp ascent and descent, stair ascent and descent. The study shows that the WaveNet, i.e., a full CNN, achieves an average F1-score of 83.0% (with SD of 0.052) in the case of one IMU, and an average F1-score of 95.58% (with SD of 0.05) in the case of two IMUs. Moreover, the WaveNet achieves a peak F1-score of 87.17% on subject AB193 (one IMU), and a peak of 97.88% on subject AB194 (two IMUs).

The potential of the present method to predict the locomotion intent of amputees and to control lower limb prostheses is left as future work.

References

[1] B. Hudgins, P. Parker, and R. N. Scott, “A new strategy for multifunction myoelectric control,” *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.

[2] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Computing Surveys*, vol. 46, no. 3, 2014.

[3] K. Zhang, C. W. de Silva, and C. Fu, “Sensor fusion for predictive control of human-prosthesis-environment dynamics in assistive walking: A survey,” *arXiv preprint arXiv:1903.07674*, 2019.

[4] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[5] J. Figueiredo, S. P. Carvalho, D. Goncalves, J. C. Moreno, and C. P. Santos, “Daily locomotion recognition and prediction: A kinematic data-based machine learning approach,” *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.

[6] A. J. Young, A. M. Simon, and L. J. Hargrove, “A training method for locomotion mode prediction using powered lower limb prostheses,” *IEEE Transactions on Biomedical Engineering*, vol. 22, no. 3, pp. 672–677, 2014.

[7] H. A. Varol, F. Sup, and M. Goldfarb, “Multiclass real-time intent recognition of a powered lower limb prosthesis,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 3, pp. 542–551, 2010.

[8] Y. D. Li and E. T. Hsiao-Wecksler, “Gait mode recognition and control for a portable-powered ankle-foot orthosis,” in *IEEE International Conference on Rehabilitation Robotics*, 2013, pp. 1–8.

[9] B. Chen, E. Zheng, and Q. Wang, “A locomotion intent prediction system based on multi-sensor fusion,” *Sensors*, vol. 14, pp. 12 349–12 369, 2014.

[10] M. A. Alsheikh, A. Selim, D. N. L. Doyle, S. Lin, and H.-P. Tan, “Deep activity recognition models with triaxial accelerometers,” in *AAAI Conference on Artificial Intelligence*, 2016, pp. 8–13.

[11] W. Xu, Y. Pang, Y. Yang, and Y. Liu, “Human activity recognition based on convolutional neural network,” in *International Conference on Pattern Recognition*, 2018, pp. 165–170.

[12] W.-H. Chen, Y.-S. Lee, C.-J. Yang, S.-Y. Chang, Y. Shih, J.-D. Sui, T.-S. Chang, and T.-Y. Shiang, “Determining motions with an IMU during level walking and slope and stair walking,” *Journal of Sports Sciences*, vol. 38, no. 1, pp. 62–69, 2020.

[13] O. Dehzangi, M. Taherisadr, and R. ChangalVala, “IMU-based gait recognition using convolutional neural networks and multi-sensor fusion,” *MDPI Sensors*, vol. 17, no. 12, p. 2735, 2017.

[14] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and T. Kechadi, “Human activity recognition with convolutional neural networks,” in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 541–552.

[15] B.-Y. Su, J. Wang, S.-Q. Liu, M. Sheng, J. Jiang, and K. Xiang, “A CNN-based method for intent recognition using inertial measurement units and intelligent lower limb prosthesis,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 5, pp. 1032–1042, 2019.

[16] B. Hu, P. C. Dixon, J. V. Jacobs, J. T. Dennerlein, and J. M. Schiffman, “Machine learning algorithms based on signals from a single wearable inertial sensor can detect surface- and age-related differences in walking,” *Journal of Biomechanics*, vol. 71, pp. 37–42, 2018.

[17] B. Hu, E. Rouse, and L. J. Hargrove, “Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals,” *Frontiers in Robotics and AI*, vol. 5, p. 14, 2018.

[18] R. Mahony, T. Hamel, and J. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[19] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm,” in *IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[20] A. Fu, “Real-time gesture pattern classification with IMU data,” 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imageNet classification,” in *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[22] A. van den Oord, S. Dieleman, H. Zen *et al.*, “WaveNet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.

[23] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] K. Cho, B. Van Merriënboer, C. Gulcehre *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[25] P. Xia, J. Hu, and Y. Peng, “EMG-based estimation of limb movement using deep learning with recurrent convolutional neural networks,” *Artificial Organs*, vol. 42, no. 5, pp. E67–E77, 2018.

[26] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv preprint arXiv:1412.6980*, 2014.