university of groningen / faculty of science and engineering

IMU-based deep neural networks: Prediction of locmotion intentions and transitions for an osseointegrated transfemoral amputee

Julian Bruinsma, s3215601, j.bruinsma.6@student.rug.nl,
Supervisor: Prof Dr R. Carloni

Abstract: This paper focuses on the design and comparison of different deep neural networks for the real-time locomotor intention prediction of one osseointegrated amputee by using data from an inertial measurement unit (IMU). The deep neural networks are based on convolutional neural networks, recurrent neural networks, and convolutional recurrent neural networks. The input to the architectures are features in both time-domain and time-frequency domain, which are derived from either one IMU placed on the upper left thigh or two IMUs placed on both the left thigh and left shank of the osseointegrated amputee. The prediction of eight and seven different locomotion modes and twenty-four and twenty transitions are investigated with or without sitting, respectively. The study shows that a recurrent network, realized with four layers of gated recurrent unit networks, achieves, with 5-fold cross-validation, a mean F1-score of 84.77% and 86.5% using one IMU and 93.06% and 89.99% using two IMUs, with or without sitting, respectively.

1 Introduction

Prostheses play an important role in the daily life of amputees. In the case of individuals with lower-limb amputations, the need to conveniently perform daily activities, such as walking, standing up, and stair climbing are important [1]. A fundamental step in developing active lower-limb prostheses is achieving intuitive control, where the locomotor intention should be accurately predicted. To avoid discomfort in the prosthetic leg and reduce the cognitive load the locomotor intention should be predicted and converted within 300 ms [2].

An inertial measurement unit (IMU) is designed to measure angular acceleration and angular velocity and has been applied in variously wearable products. A variety of data analysis and machine learning techniques have been proposed in the literature to translate information from the IMU into locomotion modes in real-time. These pattern recognition techniques can be broadly divided into two categories, namely, methods based on feature engineering [3] and methods based on feature learning [4], either with handcrafted or raw input data. Feature engineering methods have been studied in locomotion intent prediction and locomotion mode recognition using IMU. In [5] handcrafted features from the time domain were extracted to compare different supervised machine learning algorithms, i.e. support vector machine, multi-layer perceptron, random forest, k-nearest-neighbours and discriminant analysis. [6] translated raw signals to estimate translational motion of the lower leg from which time-domain features were extracted. Other research focused on handcrafted features from the fusion between IMU and other sensors, such as a pressure sensor [7].

Feature learning, using deep learning methods, has the advantage of extracting higher-level features from the data and does not rely on human experience or domain knowledge [4]. Therefore, feature learning has also been recently used in locomotion mode recognition and locomotion intent prediction. For example, a single triaxial accelerometer has been used in combination with deep belief networks [8] and convolutional neural networks (CNNs) [9], using the spectrogram or extracted time-domain features, respectively. Additionally, CNNs, in combination with one IMU on the foot [10], multiple IMUs on the lower-limb [11] and several IMUs on the lower-limbs and/or torso [12], [13] have been investigated, together with a recurrent neural network (RNN) using two IMU on the arms [14].

Table 1.1: State of the art of deep learning and machine learning methods on motion prediction of upper and lower limbs using IMU data. The upper part of the table indicates the feature engineering methods using machine learning and the lower part of the table the feature learning methods using deep learning. IMU indicates the use of both accelerometer and gyroscope data.

Ref.	Year	Method	Features	Mean accuracy	Limb(s)	Motion(s)	Transitions	Subject(s)
Feature Engineering Methods								
[5]	2020	Gaussian SVM	IMU time domain handcrafted)	97.65%	Upper, lower	Locomotion (5 modes)	Yes (9 transitions)	10 healthy
[6]	2018	LDA	IMU time domain + kinematics (handcrafted)	96.22%	Lower	Locomotion (5 modes)	No	6 transtibial amputees
[7]	2014	LDA	IMU time domain + pressure insoles (handcrafted)	99.71%	Lower	Locomotion (6 modes)	Yes (12 transitions)	7 healthy
Feature Learning Methods								
[8]	2016	DBN (5 hidden layers)	Accelerometer time-frequency domain	98.23%	Lower	6 activities	No	29 healthy
[8]	2016	DBN (5 hidden layers)	Accelerometer time-frequency domain	91.5%	Lower	Locomotion (gait freeze classif.)	No	10 patients
[8]	2016	DBN (5 hidden layers)	Accelerometer time-frequency domain	89.38%	Upper	10 activities	No	1 healthy
[9]	2018	CNN (4 hidden layers)	Accelerometer time domain	91.97%	Lower	6 activities	No	36 healthy
[10]	2020	CNN (12 hidden layers)	IMU raw data	87.74%	Lower	Locomotion (6 modes)	No	30 healthy
[11]	2019	CNN (6 hidden layers)	IMU raw data	94.15%	Lower	Locomotion (5 modes)	Yes (8 transitions)	10 healthy
[11]	2019	CNN (6 hidden layers)	IMU raw data	89.23%	Lower	Locomotion (5 modes)	Yes (8 transitions)	1 transtibial amputee
[12]	2019	CNN (7 hidden layers)	IMU raw data	NA	Lower	16 lower limb motions	No	19 healthy
[13]	2017	CNN (10 hidden layers)	IMU time-frequency domain	97.06%	Upper, lower	Locomotion (gait phase classif.)	No	10 Healthy
[14]	2018	RNN (1 hidden layer)	IMU time domain	96.63%	Upper, lower	5 activities	No	11 healthy
[15]	2019	CNN (3 hidden layers)	IMU time domain	95.58%	Lower	Locomotion (6 modes)	No	10 healthy
[16]	2017	RNN (3 hidden layers)	IMU time domain	77%	Upper	3 hand motions	No	1 healthy
[17]	2016	CNN (3 hidden layers)	IMU raw data	97.01%	Lower	locomotion (6 modes)	No	12 healthy
[18]	2019	NN (1 hidden layer)	IMU raw data	98.60%	Upper	10 hand motions	No	5 healthy

To convert the locomotion control mode smoothly, correctly, and in time it is important to predict transitions as well. Few works have been presented on predicting locomotion mode transitions in combination with IMU data [5], [7], [11]. The authors of [5] used five IMUs on different limbs to create handcrafted features for each limb within the time domain, such as the mean, standard deviation, range and first and last value of the angle within a window. For classification, a Gaussian support vector machine (SVM) was used. Although the accuracy in [5] is relatively high (97.65%) for the Gaussian support vector machine, the prediction of specific steady-state locomotion and transitions were separated. Additionally, [7] used the time domain features: maximum, minimum, mean, waveform length, standard deviation and root mean square of both the IMU and a pressure insole signals within a window. The classification was made using linear discriminant analysis (LDA). The accuracy shown in Table 1.1 is only for the steady-state locomotion. The transitions were all correctly classified but were not taken into account when reporting the accuracy.

To the best of the author's knowledge, only [11] used a deep learning method. More specifically, a CNN combined with three IMU's on the lower limbs achieved an accuracy of 89.23% for a transtibial amputee. The features used in [11] were only raw IMU signals, as opposed to the aforementioned machine learning methods.

Table 1.1 reports the main contributions of deep neural networks and machine learning networks of the state of the art methods for motor intention prediction. The table also reports the average accuracies, the inclusion of transitions, and whether the networks were tested on healthy subjects or amputees.

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 networks, based on CNN, RNN and convolutional recurrent neural networks (CRNNs) designed by [15] [19] have been adjusted and compared. The inputs into the architectures are features in the time-domain and time-frequency domain, which have been extracted from either one IMU, placed on the left thigh, or two IMU's

placed on the left thigh and shank. Specifically, the features are mean IMU data (i.e., angular accelerations and angular velocities, obtained from 3-axis accelerometers and 3-axis gyroscopes) from a time window, which is chosen to complement the frequency information within that time window, rather than having more raw data points than frequencies. Additionally, the corresponding quaternions and frequency information, which has been obtained using short-time Fourier transform on each IMU channel, have been chosen. These features are extracted and validated on one osseointegrated transfemoral amputee. The task concerns the prediction of seven or eight locomotion actions: standing, stair ascent and descent, ramp ascent and descent, terrain walking either with or without sitting and the twenty-four or twenty transitions, respectively. This study shows that a RNN, realized with four layers of gated recurrent units, achieves, with 5-fold cross-validation, a mean F1-score of 84.77% (standard deviation of 1.33) and 86.5% (standard deviation of 0.38) using one IMU and 93.06% (standard deviation of 1.21) and 89.99% (standard deviation of 5.95) using two IMUs, with or without sitting, respectively.

The remainder of the paper is organized as follows. In Section 2, the materials and methods for locomotor intention prediction are described. Section 3 will present the results, which are discussed in Section 4. The concluding remarks will be made in Section 5.

2 Materials and Methods

This Section presents the design of nine different deep neural networks that learn IMU features in both the time and time-frequency domain for real-time prediction of eight locomotion modes and twenty-four transitions.

2.1 Data-set

The data-set used in this study is provided by the MyLeg project [20] under the name "MyLeg - Amputee Pilot". The data have been collected on one male osseointegrated amputee with a weight of 84.1 kg, a height of 186.6 cm and an amputation of the left leg. The data were extracted from the subject during locomotion using wearable electromyographic (EMG) sensors and IMUs. From the data-set, this study only uses data from two IMUs on the left thigh and left shank, which is the side of the prosthetic leg. The IMU data were initially sampled with a sampling frequency of 240 Hz but were later re-sampled to 1000 Hz to synchronize with the EMG data.

The locomotion modes that need to be predicted are S: Sitting, ST: Standing, LW: Level Ground Walking, SA: Stair Ascent, SD: Stair Descent, RA: Ramp Ascent, RD: Ramp Descent and TW: Terrain Walking. The transitions that need to be predicted are shown in Table 2.1. The left column indicates the starting mode and the right column indicates the ending modes in the trial. The ramps have a slope of 10 degrees for three meters and continue on 15 degrees. The inclination of the stairs was not provided. The data labelling has been done manually and transitions were initially labelled as the future mode (i.e. S $\rightarrow$ ST was labelled as 'ST'). To include transitions in the data-set, a window of 500 ms was chosen between two modes, i.e. 250 ms in the previous mode and 250 ms in the future mode, and labelled with the correct transition label.

Table 2.1: Different transitions between modes indicated by a mode before transition and a mode after transition.

Locomotion mode before transition	Locomotion mode after transition						
S	ST	W					
ST	S	W	SA	SD	RA	RD	TW
W	S	ST	SA	SD	RA	RD	TW
SA	ST	W					
SD	ST	W					
RA	RD						
RD	W						
TW	ST	W					

2.2 Input

1) Features: The input data into the deep neural networks are extracted from the IMU data. Specifically, either one IMU on the left thigh or two IMUs on the left thigh and left shank are used. The features used in this study are the mean of the raw IMU data within a window, W, the according quaternions and time-localized frequency information of each IMU channel, calculated using the short-time Fourier transform (STFT). The number

of inputs from the STFT is related to W and are not equal to the number of raw data points from the IMU. Therefore, the mean of each channel from the IMU is taken within W, resulting in both time and time-frequency domain information within W. The window has been chosen to be 30 ms with a step length of 10 ms, to leave enough time to process data, predict the locomotion mode and convert into the right control mode [2]. The quaternions are estimated using the mean IMU data points by the filter proposed in [21], with the implementation of [22].

2) Sequential frames: The input data is generated using a window of 30 ms. Five adjacent samples are sequentially concatenated into one frame using a sliding window with stride 1, which equals to an overlap of 20 ms, resulting in a frame of 70 ms. Figure 2.1 shows how frames are extracted from the data set.

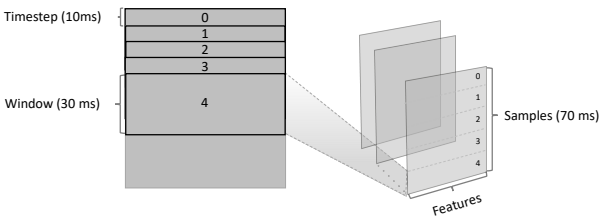

Figure 2.1: Representation of sequential frames. Using a window of 30 ms and time-step of 10 ms the samples were sequentially concatenated.

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

4) Data partitioning: Using 5-fold cross-validation, 80% of the data was used for training and 20% was used for testing. Within training, 10% was used for validation.

2.3 Output

The output of the neural networks has a dimension equal to the locomotion modes to be predicted: standing, ground-level walking, stair ascent and descent, ramp ascent and descent terrain walking and sitting, and the transitions (see Table 2.1). Consequently, the exact output dimension is thirty-two when sitting is included and twenty-seven when sitting is excluded.

2.4 Deep Neural Networks

Nine deep neural networks were designed and compared in this research. The architectures are further described in the following subsections. The networks are mainly based on CNNs, RNNs and CRNNs and are inspired by [15] and [19].

2.4.1 Convolutional Neural Networks

Three different CNN architectures (i.e. CNN1D, CNN2D and WaveNet) have been designed. Figure 2.2 shows both the CNN1D and the CNN2D architectures, which consist of six hidden layers i.e. four convolution layers and two dense layers. The input into the networks consists out of five rows, i.e. five concatenated samples and the number of features (see Figure 2.1). In this study, the convolutional kernel size is set to (1×2) and (2×2) for the CNN1D and CNN2D, respectively. The first four convolutional layers have a filter size of respectively 32, 64, 128 and 256. A rectified linear unit (ReLu) is used as an activation function in each filter. Finally, two dense layers follow: first a dense layer with 50 units and a dropout of 0.25 and then an output layer that has the units equal to the number of classes (thirty-two or twenty-seven) and a Softmax activation function. The most significant difference between the CNN1D and CNN2D the direction of the convolution kernels. CNN1D slides only frame-wise, i.e. from top to down, while CNN2D slides both frame-wise and column-wise.

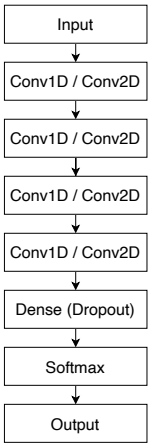

Figure 2.2: The CNN1D and CNN2D architectures consist of six hidden layers, including four convolutional layers and two dense layers.

Figure 2.3 shows another CNN, the WaveNet [23], which consists of four convolutional layers. The input is first processed by a causal convolutional layer, consisting out of 256 filters and a filter size of 2. Next, the output goes in two ways. In one direction it is served as an input into a dilated convolutional layer (256 filters and a filter size of 2), which consists out of two convolutions with either a tanh or sigmoid activation function, and these are combined using dot multiplication. This then goes to the output layer. In another direction, it skips the dilated convolution and is directly summed up with the output of the dilated convolution, which then serves as an input for the second layer. The output layer consists of a dense layer (200 units, 0.25 dropout) and a dense layer with a Softmax activation function, where the number of units is equal to the number of classes.

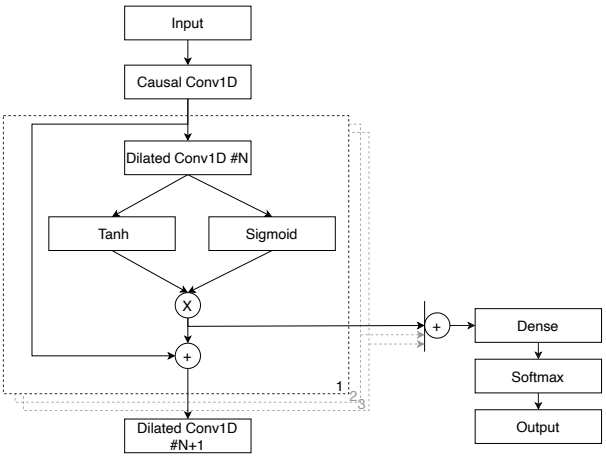

Figure 2.3: WaveNet architecture with four convolutional layers, from which three are dilated and one is causal, and two dense layers.

2.4.2 Recurrent Neural Networks

Figure 2.4 shows two different RNN architectures that have been designed. Both architectures consist of six hidden layers, i.e. four recurrent layers and two dense layers. The recurrent layers can either be long short-term memory (LSTM) [24] or gated recurrent units (GRU) [25]. The first four layers consist out of 128 LSTM or GRU units. Then two dense layers follow, one that has 200 units and a dropout of 0.25 and one that serves as an output layer with units equal to the number of classes.

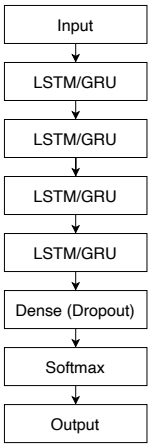

Figure 2.4: RNN architectures with four recurrent layer, either LSTM or GRU, and two dense layers.

2.4.3 Convolutional Recurrent Neural Networks

Figure 2.5 shows the four different CRNN networks that have been designed. They both consist out of eight hidden layers, i.e., three convolutional layers (either one- or two-dimensional), three recurrent layers (either LSTM or GRU), and two dense layers. The first three convolutional layers are similar to that of the CNNs with the only difference that it has filters of size 64, 128 and 256, respectively. The last three RNN layers are equivalent to the layers in the RNN in Figure 2.4, as they also have 128 units. The final two layers are both dense layers. One of them has 200 units and 0.25 dropout and the other has a number of units equal to the number of classes and a Softmax activation function. There is only one difference between the one-dimensional and two-dimensional version of the CRNN and that is that the output of the CNN2D layers needs to be wrapped together with the time-step to serve as a compatible input into the RNN layers.

2.5 Hyperparameters

This section describes the parameters that were set for the training procedure. The training was done on a single computer with an NVIDIA GeForce GTX 1060, a quad-core Intel i7-6700 processor and 8 GB RAM.

1) Learning Rate: The learning rate is set at a

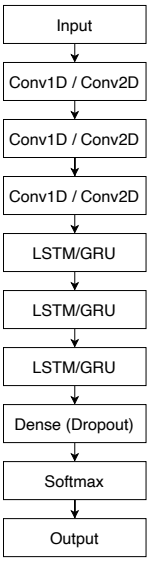

Figure 2.5: CRNN architectures with three convolutional layers (one- or two-dimensional), three recurrent layers (LSTM or GRU) and two dense layers.

value of 0.001. The trade-off value is determined at such, because using a high learning rate causes the network to never converge, while a lower learning rate would increase the risk of falling into a local minimum.

2) Optimizer: The optimizer is chosen to speed up the convergence of the neural network, by optimizing the gradient descent. The Adaptive Moment Estimation (Adam) has been used in this study [26]. Adam computes individual adaptive learning rates for different parameters.

3) Batch Size: The batch size represents how many input data are show simultaneously to the network before updating its weights. The batch size is chosen to be 512, which is relatively high, but in combination with the number of filters, it has been observed to obtain the highest performance. The high batch size does not result in processing more previous data before classifying a mode, as the data is shuffled, but may increase the accuracy of the error estimate in training.

4) Loss Function: As a loss function categorical cross-entropy has been used.

5) Class Weighting: The loss function assumes there is an equal distribution among the different classes. However, in this study, the transition classes are underrepresented. To account for this unbalance a weight is added to each class. This weight makes the transitions more important for the network and penalizes mistakes made for the transitions more opposed to other classes.

6) Shuffling: The training of a network is done by feeding the network the input data batch by batch. If the network is fed the data in chronological order the network would overfit between multiple classes, because the data is first on solely the first mode and then the next mode. To avoid this the sequential frames are shuffled. Thus, the distribution of classes in a batch is more equivalent to the distribution of the data-set, but the temporal properties of the data remain within the frame.

7) Epoch: The data is presented 150 times to the networks during training to optimize data use and avoid under-fitting.

8) Early stopping: The number of epochs is set to a high number to ensure that the data is used enough and is not under-fitting. If the number of epochs is too high the network will start to overfit on the data. Therefore, an early stopping approach is used. If the accuracy on the validation set has not increased for fifteen epochs the network will stop training and will be the final model. This number is empirically set to avoid an increase in validation loss, which is a sign of overfitting.

2.6 Evaluation: Performance metric

Due to the uneven distribution of the data the neural networks are compared based on the F1 scores, a metric that compares both precision and recall and is calculated using:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

where

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

with tp being the number of true positive predictions, fp the number of false positives and fn the number of false negatives. The final comparison of the networks is based on this metric.

K-fold cross-validation is used to compare the general effectiveness of the neural networks. In this research, k is set to 5, which means that the data is

divided into five subsets. Next, the data is validated on one subset and trained on the remaining four. The validation is done on every subset and hence, happens five times in total. Finally, the mean F1 score of all validations is taken to compare the performances. This way, the data gets fully utilized and prevents underfitting and increasing the reliability of the evaluation as the training and testing is set differently each time.

3 Results

In this section, the proposed architectures are compared using the F1-score performance metric. The results are reported separately based upon the features extracted from the relative sensor placement for either one IMU on the left thigh or two IMUs on the left thigh and shank of one osseointegrated amputee. Additionally, within each subsection, the results for including or excluding the sitting 'mode' are reported separately.

3.1 One IMU (left thigh)

Figure 3.1 shows the F1-scores (mean and standard deviation (SD)), with 5-fold cross-validation of all the deep neural networks when only features from the left thigh are used for the prediction of the locomotion modes and transitions, including the sitting mode. It can be observed that the GRU outperforms the other networks with a mean F1-score of 84.77% ($SD = 1.33$). A paired t-test shows that there is a significant difference between the GRU and the LSTM (i.e. the second-best network) ($p = 0.091 < 0.05$).

Figure 3.2 shows the F1-scores, with 5-fold cross-validation of all the deep neural networks when only features from the left thigh are used, but the sitting mode and corresponding transitions are excluded from the data set. The GRU outperforms the other networks with a mean of 86.5% ($SD = 0.38$) and a paired t-test shows that the GRU has a significant difference concerning the LSTM ($p = 0.012 < 0.05$).

3.2 Two IMUs (left thigh and shank)

Figure 3.3 shows the F1-scores, with 5-fold cross-validation of all the deep neural networks when the features from both the left thigh and shank are used

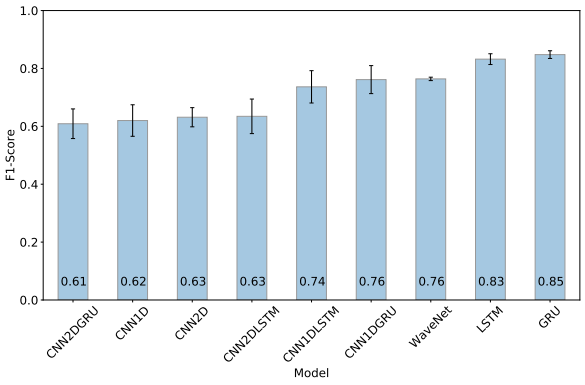

Figure 3.1: F1-score (mean and SD), with 5-fold cross validation for all the deep neural network architectures of one osseointegrated amputee. Only features from the upper left thigh were used.

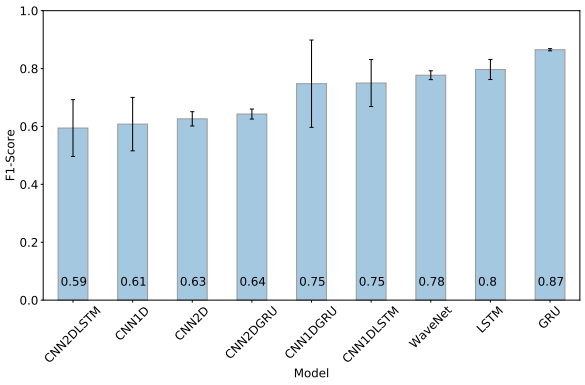

Figure 3.2: F1-score (mean and SD), with 5-fold cross validation for all the deep neural network architectures of one osseointegrated amputee. Only features from the upper left thigh were used. Sitting and corresponding transitions were excluded from the data-set.

and the sitting mode is included. Again, it seems that GRU outperforms the other networks with a mean of 93.06% ($SD = 1.21$). Although, with respect to the LSTM, the GRU has no significant difference ($p = 0.17 > 0.05\%$). However, there is a significant difference between the GRU and the WaveNet ($p = 0.006 < 0.05$).

Figure 3.4 shows the F1-scores, with 5-fold cross-validation of all the deep neural networks when the features from both the left thigh and shank are

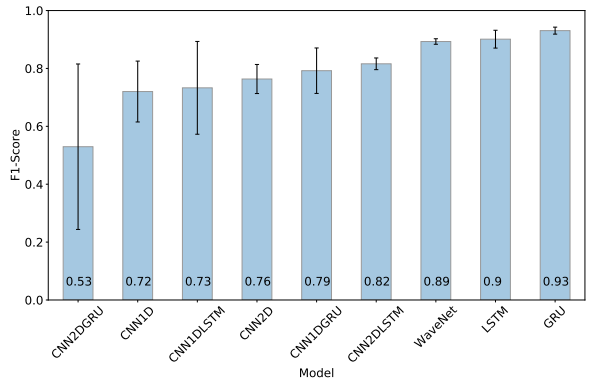

Figure 3.3: F1-score (mean and SD), with 5-fold cross validation for all the deep neural network architectures of one osseointegrated amputee. Features from the left thigh and shank were used.

used, but the sitting mode and corresponding transitions are excluded. It can be observed that the WaveNet, GRU and LSTM have seemingly similar performances. Moreover, a paired t-test indicates no significant difference between the WaveNet (mean is 90.14% and SD is 1.15) and the GRU (mean is 89.98% and SD is 5.95) ($p = 0.958 > 0.05$). Additionally, there is no significant difference between the LSTM (mean is 89.68% and SD is 1.82) and the GRU ($p = 0.989 > 0.05$).

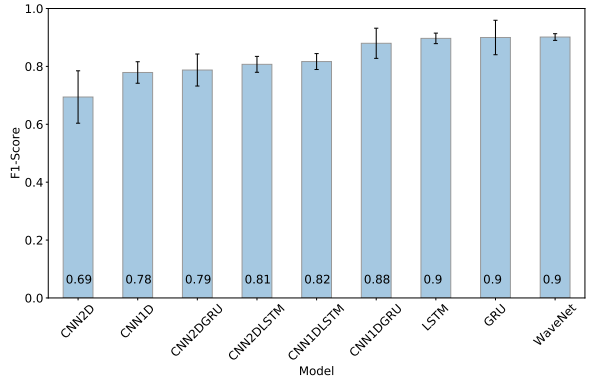

Figure 3.4: F1-score (mean and SD), with 5-fold cross validation for all the deep neural network architectures of one osseointegrated amputee. Features from the left thigh and shank were used. The sitting mode and corresponding were excluded from the data-set.

3.3 Running time

Table 3.1 shows the running time in of the three outperforming models when individually classifying 1000 samples. The samples were randomly chosen from the data set. It can be noted that the processing time does not increase when two IMUs are used, while the performance does increase significantly. Besides, the table shows that the SD of the GRU is higher than the other two networks.

Table 3.1: Mean running time and SD (in ms) of the three outperforming models when individually classifying 1000 samples (including sitting).

IMUs	WaveNet	LSTM	GRU
Thigh	17.56 ± 5.33	17.29 ± 6.02	17.79 ± 13.59
Thigh and Shank	16.87 ± 5.00	17.33 ± 6.58	18.17 ± 13.72

3.4 Results Summary

Table 3.2 summarizes the results of the four experimental conditions, i.e., one or two IMUs and with or without the sitting mode, and shows the results of the three best performing models. It can be noted that the GRU outperforms the other models except when sitting is removed from the data set and the information of two IMUs is used. However, the running time of the GRU is higher and fluctuates more between different samples than the other two networks, but remains below 300 ms.

4 Discussion

This study presented a comparison of nine different deep neural network architectures for the prediction of locomotion. These networks used inputs extracted from either one IMU or two IMUs, attached to the amputated side of an osseointegrated amputee. The mean signal, quaternions and frequency information were used as input into the networks. These were all extracted using a 30 ms window with a time-step of 10 ms and concatenated into a frame.

With a mean F1-score of 93.06%, using inputs from two IMUs (left thigh and shank), the designed GRU has a performance that is higher than in [11] for an amputee, while the decision space is over two times larger (thirty-two vs. thirteen motor intents) and less IMUs are used (two vs. three, respectively). However, compared to [11], a more complex

Table 3.2: Summary of the best performing models' results in different experimental conditions: one IMU (on the left thigh) or two IMUs (on the left thigh and shank) and with sitting in- or excluded. The highest F1-scores for each setting are bold.

Locomotor intention prediction for one osseointegrated amputee						
Features	Raw IMU data, quaternions and frequency information					
Number of features	106 (for one IMU) and 212 (for two IMUs), i.e., for each IMU: • 6 mean IMU data, • 4 quaternions estimated from mean data, and • 96 frequencies derived from short-time Fourier transform using a window of 30 ms are used					
Number of samples	76420					
Number of frames per sample	5					
Number of classes (i.e. modes and transitions)	32 (with sitting) or 27 (without sitting)					
Mean F1-score (one IMU) from 5-fold cross-validation	With sitting:			Without sitting:		
	WaveNet:	76.39%,	$SD = 0.59$	WaveNet:	77.70%,	$SD = 1.53$
	LSTM:	83.21%,	$SD = 1.86$	LSTM:	79.68%,	$SD = 3.48$
	GRU:	**84.77%**,	$SD = 1.33$	GRU:	**86.50%**,	$SD = 0.38$
Mean F1-score (two IMUs) from 5-fold cross-validation	With sitting:			Without sitting:		
	WaveNet:	89.30%,	$SD = 0.95$	WaveNet:	**90.14%**,	$SD = 1.15$
	LSTM:	90.11%,	$SD = 3.08$	LSTM:	89.68%,	$SD = 1.82$
	GRU:	**93.06%**,	$SD = 1.21$	GRU:	89.99%,	$SD = 5.95$

input is used in this study, which could mean that the computational cost is higher in the presented networks. Additionally, the designed GRU outperforms CNNs where one IMU is used [10], RNNs where two IMUs are used [14] and DBNs that use the frequency information of one IMU's accelerometer [8].

Although Table 1.1 shows higher accuracies in other research, it must be noted that the number of different locomotion modes and transitions are significantly higher in this study. Consequently, it could be assumed that performance might increase when the number of locomotion modes is decreased. However, Table 3.2 shows that the F1-score is not significantly higher in the experimental conditions where sitting is removed. Moreover, in the case of using two IMUs, the F1-score is even lower.

5 Conclusion

This paper presented a comparison of nine different deep neural networks, inspired by the work of [15] and [19], for the real-time prediction of locomotor intention. The inputs to the architecture are features from the time-domain, i.e. mean IMU data and quaternions from a 30 ms window, and features from the time-frequency domain which have been obtained using short-time Fourier transform on each IMU channel. The features were derived from either one IMU (on the left thigh) or two IMUs (on the left thigh and shank) of one osseointegrated amputee. The architectures have to predict: i) eight locomotion modes: sitting, standing, ground-level walking, stair ascent and descent, ramp ascent and descent and the twenty-four transitions between the modes; ii) seven locomotion modes and twenty transitions (i.e. sitting and its transitions are removed from the data-set)

The study shows that the RNN with four layers of gated recurrent units outperforms the other architectures in three out of four scenarios. Using one IMU it achieves a mean F1-score of 84.77% (with SD of 1.33) and 86.5% (with SD of 0.38) with and without sitting, respectively. Using two IMUs it achieves a mean F1-score of 93.06% (with SD of 1.21) and 89.99% (with SD of 5.95) with and without sitting, respectively. These performances were achieved by taking the mean of a 5-fold cross-

validation on one subject.

References

[1] H. Pernot, L. De Witte, E. Lindeman, and J. Cluitmans, "Daily functioning of the lower extremity amputee: An overview of the literature," *Clinical Rehabilitation*, vol. 11, no. 2, pp. 93–106, 1997.

[2] B. , 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.

[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. Gonçalve, J. C. Moreno, and C. P. Santos, "Daily locomotion recognition and prediction: A kinematic data-based machine learning approach," *IEEE Access*, vol. 8, pp. 33 250–33 262, 2020.

[6] R. Stolyarov, G. Burnett, and H. Herr, "Translational motion tracking of leg joints for enhanced prediction of walking tasks," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 4, pp. 763–769, 2017.

[7] B. Chen, E. Zheng, and Q. Wang, "A locomotion intent prediction system based on multi-sensor fusion," *Sensors*, vol. 14, no. 7, pp. 12 349–12 369, 2014.

[8] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[9] W. Xu, Y. Pang, Y. Yang, and Y. Liu, "Human activity recognition based on convolutional neural network," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 165–170.

[10] 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.

[11] 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.

[12] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and T. Kechadi, "Human activity recognition with convolutional neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 541–552.

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

[14] R. R. Drumond, B. A. Marques, C. N. Vasconcelos, and E. Clua, "Peek - An LSTM recurrent network for motion classification from sparse data." in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Application*, vol. 1, 2018, pp. 215–222.

[15] H. Lu, L. R. B. Schomaker, and R. Carloni, "IMU-based deep neural networks for locomotor intention prediction," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[16] A. Fu and Y. Yu, "Real-time gesture pattern classification with IMU data," 2017.

[17] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Human activity recognition with inertial sensors using a deep learning approach," in *2016 IEEE SENSORS*. IEEE, 2016, pp. 1–3.

[18] M. Kim, J. Cho, S. Lee, and Y. Jung, "IMU sensor-based hand gesture recognition for human-machine interfaces," *Sensors*, vol. 19, no. 18, p. 3827, 2019.

[19] H. Lu, "A lower limb specialized motor intent recognition system using neural networks," Master's thesis, University of Groningen, 2019.

[20] (2018). [Online]. Available: http://myleg.eu

[21] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on automatic control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[22] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE international conference on rehabilitation robotics*. IEEE, 2011, pp. 1–7.

[23] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[24] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "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. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.