Cognitive Computation
https://doi.org/10.1007/s12559-020-09754-0

Echo State Networks and Long Short-Term Memory for Continuous Gesture Recognition: a Comparative Study

Doreen Jirak[1] · Stephan Tietz[2] · Hassan Ali[1] · Stefan Wermter[1]

Received: 20 February 2020 / Accepted: 15 July 2020

Abstract

Recent developments of sensors that allow tracking of human movements and gestures enable rapid progress of applications in domains like medical rehabilitation or robotic control. Especially the inertial measurement unit (IMU) is an excellent device for real-time scenarios as it rapidly delivers data input. Therefore, a computational model must be able to learn gesture sequences in a fast yet robust way. We recently introduced an echo state network (ESN) framework for continuous gesture recognition (Tietz et al., 2019) including novel approaches for gesture spotting, i.e., the automatic detection of the start and end phase of a gesture. Although our results showed good classification performance, we identified significant factors which also negatively impact the performance like subgestures and gesture variability. To address these issues, we include experiments with Long Short-Term Memory (LSTM) networks, which is a state-of-the-art model for sequence processing, to compare the obtained results with our framework and to evaluate their robustness regarding pitfalls in the recognition process. In this study, we analyze the two conceptually different approaches processing continuous, variable-length gesture sequences, which shows interesting results comparing the distinct gesture accomplishments. In addition, our results demonstrate that our ESN framework achieves comparably good performance as the LSTM network but has significantly lower training times. We conclude from the present work that ESNs are viable models for continuous gesture recognition delivering reasonable performance for applications requiring real-time performance as in robotic or rehabilitation tasks. From our discussion of this comparative study, we suggest prospective improvements on both the experimental and network architecture level.

Keywords Continuous gesture recognition · Echo state networks · Long Short-Term Memory

Introduction

Continuous gesture recognition is a challenging task due to three critical aspects: (1) the correct identification of the start and end of the actual gesture, called *subgesture*, (2) the recognition of a gesture of possibly variable length, also called *inter-subject variability*, and (3) the accurate distinction between an active gesture and subtle movements or silent phases like pauses. The correct yet fast recognition of gestures is an important research area predominantly for vision-based application in human-robot interaction (HRI) or human-computer interaction (HCI). Although visual gesture recognition allows the most intuitive interface between a human and an agent, it is also the most challenging task starting from the recording procedures to preprocessing of a huge number of video streams to finally computational models with low latency and high recognition rates. In recent years, deep learning techniques emerged as a new way to learn huge datasets using GPU computing. Especially for gesture recognition, they achieved high accuracy on benchmarks like ChaLearn [7]. Learning sequences demands some memory mechanism as is implemented in recurrent neural networks (RNNs). Training of deep models, i.e., network architectures with many layers, through gradient propagation often suffers from effects of exploding or vanishing gradients [3, 4]. To address this issue, Long Short-Term Memory (LSTM) networks [13] have been proposed, whose gating mechanisms integrated into an RNN architecture overcome

✉ Doreen Jirak
jirak@informatik.uni-hamburg.de

[1] Department of Informatics, Knowledge Technology, University of Hamburg, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

[2] Technical University of Berlin, Strasse des 17. Juni 135, 10623 Berlin, Germany

Published online: 07 October 2020

the error-prone gradient computations. An alternative paradigm to the traditional RNN training subsumed under the term "Reservoir Computing" (RC) [26] has become popular and showed high performance in time-series prediction. A special implementation called echo state networks (ESNs), proposed by Jäger [14], has been successfully applied to language processing [12, 25], navigation tasks [6] and central pattern generation [28]. The RC community is also growing in the recent years due to the successful implementation of reservoirs in hardware [2, 23], supporting real-world applications like human action recognition [1]. Although gestures are sequences similar to sentences, human actions, or path trajectories, surprisingly little is known about the potential application of ESNs to the task of gesture recognition [8, 15].

In this article, we present an extension of our previous study on sensor-based continuous gesture recognition [24]. Although this research is dominated by vision data, other important areas like rehabilitation, limb prosthesis or controlling virtual environments use sensors like the so-called "inertial measurement unit" (IMU). The reason is that this sensor delivers movement data for direct input very fast which allows real-time control for hand or arm movements or instantaneous reactions in robotic interfaces.

Our paper is structured as follows: we will first review recent research on smart devices and the different learning techniques for continuous gesture recognition. In the subsequent section, we will summarize our ESN framework including the data recordings introduced earlier [24], followed by the explanations of our new experiments using an LSTM network. The performance of both approaches will be compared in the evaluation section and contrasted with other approaches in our discussion. We conclude our paper with suggestions for prospective applications.

Related Work

Wearable or smart devices have influenced different research domains such as controlling games and media applications or prostheses and rehabilitation. Recent work on continuous gesture recognition using smart sensors primarily uses a set of standard learning techniques such as dynamic time warping and only a few studies apply recurrent neural networks to the task. Gupta et al. [10] introduced an algorithm, which maps the sensory stream from the gyroscope and accelerometer of a Samsung mobile phone into a gesture codebook. To distinguish between an actual gesture and no gestural activation, the dynamic time warping (DTW) algorithm was used. Basically, the DTW procedure aligns two sequences which may vary in speed and, based on predefined similarity measures, can classify a sequence to its most similar sequence or "template" in a data corpus. On a set of 6, respectively, 12 gestures (actually mirrored), their approach achieved an average accuracy of 90% and 94% for users using the so-called portrait mode. Interestingly, the performance dropped significantly compared with uWave [19], a gesture recognition system for accelerometer data introduced earlier. However, due to the lack of benchmark data, a fair comparison between different systems is difficult. Although the authors demonstrated good performance with a rather simple approach, they fall short of the number of gestures and it remains open whether the creation of a codebook would scale up when extending the gesture vocabulary.

Yang et al. [29] presented a system using data from a surface electromyography sensor (sEMG) on an arm. A sliding window procedure was used to segment the sensor stream and a threshold applied to separate active gestures from unintentional gestures or noise. To model the gestures, Gaussian Mixture Models and Hidden Markov Models (GMM; HMM) were trained and the divergence between two models used for evaluation. The Kullback-Leibler divergence displayed the difference between any two models and was used to distinguish the 6 gesture classes. The system achieved 97–100% accuracy, however, the whole procedure was tested on samples from one person only. Also, the performance time for the gestures was always set to 4 seconds, which means that the models neither captured any variances between users nor gesture performance time. This aspect lowers the generalization to other users and limits the system application. Furthermore, the chosen model suite is known to be hard to train and thus, again, the question of whether the system would scale well to the number of gestures remains open.

In the context of HCI, wearables are also used in the gaming domain because the sensor input directly measured from the subject allows real-time processing. In this regard, Li et al. [18] presented a system to control the Jump&Go Fly Bird game. Gyroscope and accelerometer data were collected from a wristband and gestures manually segmented using video information to mark the start and end phase of a gesture. A sliding window approach in combination with DTW classified the game control gesture, which was limited to raising a hand. All other gestures were subsumed as "other" gesture. Although the system achieved an F1 score of up to 99%, the application of this gesture recognition system is restricted to binary classification. Moreover, manual labeling becomes a time factor when increasing the gesture vocabulary, thus more sophisticated methods for gesture spotting need to be developed.

One such system presenting the application of recurrent neural networks (RNNs) was introduced in the SLOTH architecture [5]. A triaxial wearable accelerometer provided data for 6 defined gestures classes and, additionally, a "no gesture" class. The gesture spotting was trained

with a Long Short-Term Memory (LSTM) network. A subsequent continuous gesture recognition module (CGR) then classified the different gestures using two sets: one dataset comprised sequences from 9 participants with a total of 540 sequences, evaluated offline yielding an accuracy of 96.9%. A second dataset was restricted to one participant only with combinations of different gestures. The CGR module then classified the incoming data stream in an online fashion, which substantially decreased the recall of gestures while still yielding an average accuracy of 79.7% up to 90.6% when changing some critical system parameters. Especially the "no gesture" class was misclassified 100%, which the authors [5] explained by the accelerometer settings in the device. The differences in the recall for the other gestures may also be subject to the so-called *subgesture* problem, i.e., an online classifier may output the incorrect label because it confuses similar start patterns of different gestures.

Sosin et al. [22] included domain adaptation in their system to enhance the generalization of gesture recognition to other persons. An sEMG sensor tracked hand movements from 5 subjects in two conditions, mobile and immobile wrist. An additional Leap motion device indicated the correct position of the hand. The authors compared simple and gated recurrent neural units and, additionally, trained the RNNs with adversarial domain adaptation (ADA). Employing ADA can help to prevent overfitting, and thus, a trained network can be transferred to test different subjects. The study revealed the superior performance of the simple units in combination with ADA for both conditions, evaluated using the (normalized) root mean-squared error, which measures the displacement between the angle of the correct gesture to the predicted one. The system is, however, limited by the installation of the Leap motion controller, which needs additional calibration for every new user.

Similarly, Han and Yoon [11] proposed a system for the recognition of 6 gestures obtained from a wireless triaxial gyroscope worn by 5 subjects. The gyro data was preprocessed using a sliding window and the normalized covariance between the sequences calculated. The maximum covariance between different gestures was then assigned to as the correct gesture and compared with a reference vector from a trained support vector machine (SVM). The evaluation of the single gestures performed by 4 of the participants showed an average accuracy of 97% where the confusion could be tracked back to symmetric gestures like "left-right" or "up-down." However, the system was optimized to every user with a preceding customization session. The evaluation is, therefore, biased, as the session familiarizes each subject with the task resulting in a stable gesture performance yielding low variances among the gesture classes and clear gesture waveforms. Regarding the symmetric gestures, the system also showed no improvement when trained specifically for a multimedia application.

A recent study by Wang and Ma [27] similarly to the one presented in this paper introduced a recognition system for 10 gestures performed by 40 users. The IMU from a wearable sensor was used, yielding six features that were corrected for unit differences and further projected to a lower-dimensional space by principal component analysis. The continuous gestures were segmented with a sliding window and manually labeled. An SVM was trained to output class labels, while a DTW enhanced the correct recognition for variable-length sequences. The experiments were divided into three scenarios: the recognition of a single gesture, a sequence of different but predefined gestures, and the recognition of arbitrary gesture combinations. The single gesture condition achieved an average accuracy of 93.14% and when including a timing threshold, 97.28%, mostly confusing a "circle" gesture or "up-down." However, single gestures are hardly relevant to the task of recognizing gestures in a continuous stream. For the two other experiments, the performance dropped to 86% for a predefined sequence of gestures and decreased even more when considering a random gesture sequence to 60%. This result confirms that continuous gesture recognition is a nontrivial task, heavily influenced by *inter-subject variability* in gesture performance.

Dataset and Methodology

We created our own dataset because we are using sensor-based data for gesture recognition and no public dataset is available for this task. Therefore, we make our data and the code of our models publicly available[1]. We also explain the experimental settings for both architectures used in this study: an echo state network (ESN) and a Long Short-Term Memory (LSTM) network.

Dataset Creation

First, we defined 10 gesture classes shown in Fig. 1 inspired by Lee et al. [17]. We chose the gesture types we think correspond best to an action alternatively to swiping or speech commands. For instance, the first two *snap* gestures in Fig. 1 can be used to slide photos to the left or right, or to control the volume of a music application. Second, we set up an experimental environment as demonstrated in Fig. 2 and invited 5 participants who performed each gesture 10 times, resulting in a total of 500 variable-length sequences. The input data was collected from the inertial measurement unit (IMU) of a smartphone with Android OS. The IMU

[1] https://github.com/swtietz/UHH-IMU-gestures-comparison

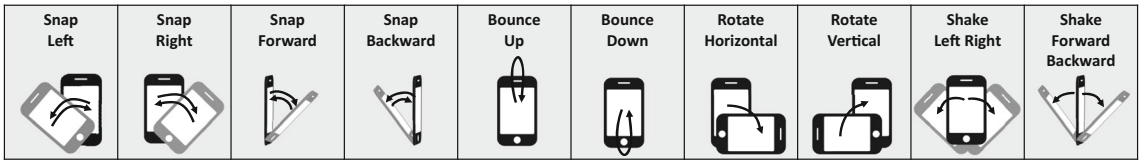

Fig. 1 We defined ten gestures used for our experiments. The challenge for a system is to recognize variable-length gestures (cf. *snap* vs. *shake*), where one shorter gesture may be a subgesture of another, and to distinguish gesture symmetry (e.g., up vs. down). (Figure from [24])

delivers 9 features: the three orientation axes $\{x, y, z\}$, the rotation velocity along these axes, and, correspondingly, the acceleration. Before feeding the sensor values into the network they are normalized channel wise, such that the maximum norm of each three dimensional signal is 1. The normalization has been carried out over the whole dataset. The signals have a frequency of 30 Hz, i.e., one time step corresponds to ≈ 0.03 s. Figure 2 outlines the data recordings: a participant was seated on a table opposite to a supervisor, who tracked each gesture performance and marked both the gesture onset and the gesture finish. Each participant was instructed on how to hold the phone and which gesture in which direction is expected for each trial (e.g., *shake left-right*). However, no time restrictions nor

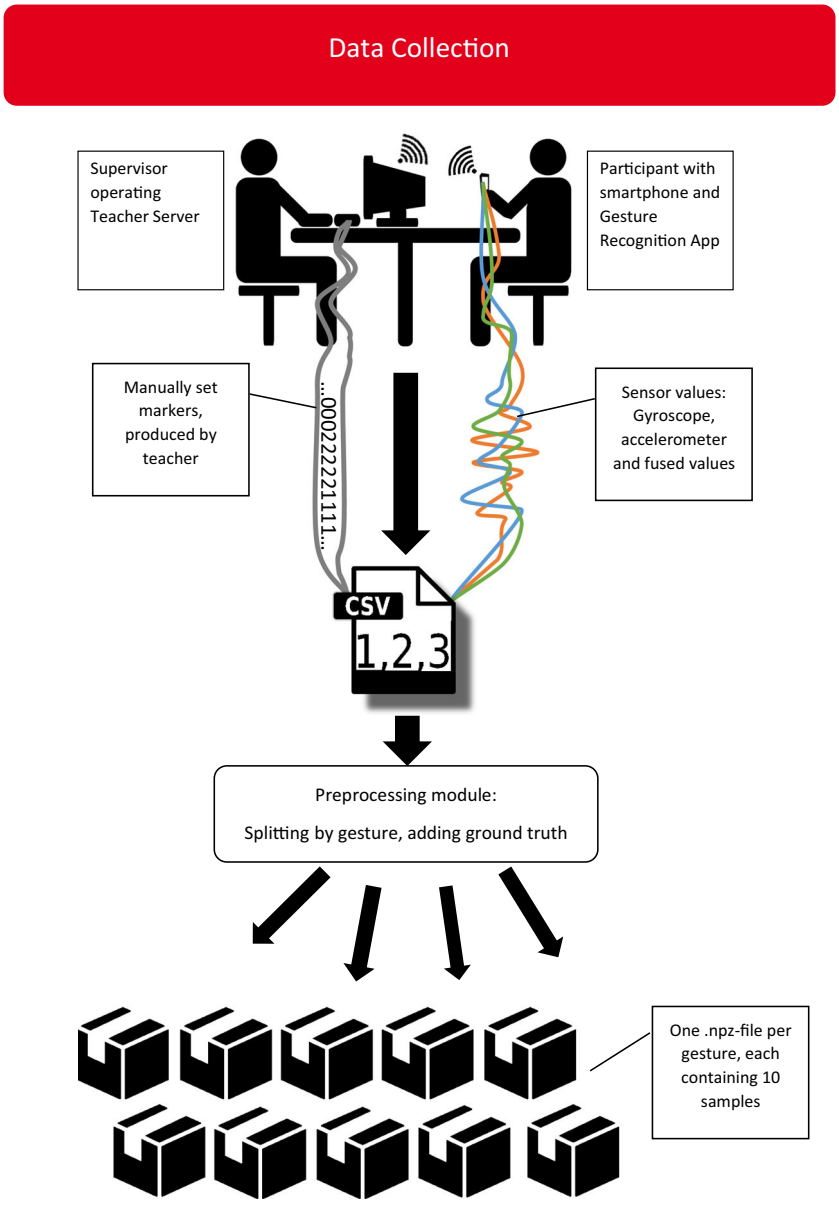

Fig. 2 The settings for the data recordings. A subject was seated opposite to the supervisor who gave instructions on when to lift the phone and which gesture type is expected. No time constraints or any further help on the gesture performance was given

help was given. After the performance of all trials per gesture type, the participants had a little break to avoid fatigue. After the recordings, the ground truth was added (cf. [24]).

All experiments follow the leave one out cross validation (LOOCV) protocol presented in our previous study [24]: we use data from $n-1$ subjects as the training set, where streams are segmented into individual gestures, concatenated to continuous sequences and shuffled randomly. Data from the remaining person was then used as the test set. We used grid search to obtain optimal parameters for both architectures. We will now explain the specific configurations of the two networks we used in this study.

Experimental Settings for the ESN

An echo state network (ESN) [14] is a specific implementation of the Reservoir Computing paradigm [26]. The model is separated into a randomly initialized hidden layer or "reservoir," which stays fixed, and a trainable readout. The key idea is to project any input to the reservoir into a high-dimensional feature space similar to kernels used by support vector machines. The projection allows the application of simpler training techniques, usually linear models. Given an input u, the reservoir states x are computed as:

$$\tilde{x}^{(t+1)} = f(u^{(t+1)}W_{in} + x^{(t)}W_{res} + y^{(t)}W_{fb} + \nu^{(t)}) \quad (1)$$

$$x^{(t+1)} = (1-\alpha)x^{(t)} + \alpha\tilde{x}^{(t+1)} \quad (2)$$

where f is the activation function (here $tanh$) and α is the leak rate. The layer-wise connectivity matrices W_* for the input, reservoir, and feedback remain fixed while training the network. We initialize W_{in} sparsely with only 10% of all weights set. Inputs are multiplied with the input scaling parameter before being fed to W_{in}. W_{res} is initialized fully connected with Gaussian weights and then re-scaled to the desired spectral radius. As we are using the ESN for supervised learning, we set the feedback matrix W_{fb} and the noise term ν to 0. We also used the full training sequences, i.e., no states were discarded. Given the teacher signal Y and the state matrix X with all states collected in matrix X:

$$Y = g(W_{out}X) \quad (3)$$

where g is the linear output activation function, the output weights W_{out} can be computed using ridge regression:

$$W_{out} = YX^T(XX^T + \lambda\mathbf{I})^{-1} \quad (4)$$

where λ is the regularization coefficient and $\mathbf{I}$ is the identity matrix. The general ESN architecture is shown in Fig. 3. The ESN used in this study has a 9 dimensional input layer, which corresponds to the sensor values obtained from the data collection. Additionally, we employed input scaling to exploit the range of the used tanh activation function [20], i.e., all input values are multiplied with a certain scaling coefficient. We fixed the reservoir to 400 neurons achieving the best performance, after having tested sizes starting from 25 neurons, doubling each results, up to 3600. We stopped our trial experiments for the reservoir size at this value as the performance started to decrease significantly. We observed an plateau-like behavior in the performance for 400, 800, and 1600 neurons, which is why we agreed on a smaller reservoir size for faster training. The output layer consists of 10 neurons, representing the 10 gesture classes. We trained the ESN using ridge regression. All hyperparameters are summarized in Table 1 with the best value resulting from our grid search in bold.

Experimental Settings for the LSTM

An LSTM [13] is an RNN capable of learning both short-term and long-term dependencies. The core concept of LSTMs lies in the cell state, whose information is regulated through three gates. First, an input gate controls which values to be stored in the cell state. Which information to be removed is then decided by a forget gate. Finally, an output gate is responsible for choosing which values to be used for the activation of an LSTM unit. Figure 4 shows the LSTM architecture used in this study. The calculations for LSTM training are shown in Eqs. 5–7 for the gate mechanisms. Equations 8 and 9 show the calculations within the layers, i.e., the cell states and the hidden states as used in our implementation.

$$i^{(t)} = \sigma(W_{ix}x^{(t)} + W_{ih}h^{(t-1)} + b_i) \quad (5)$$

$$f^{(t)} = \sigma(W_{fx}x^{(t)} + W_{fh}h^{(t-1)} + b_f) \quad (6)$$

$$o^{(t)} = \sigma(W_{ox}x^{(t)} + W_{oh}h^{(t-1)} + b_o) \quad (7)$$

$$c^{(t)} = f^{(t)} \circ c^{(t-1)} + i^{(t)} \circ \tanh(W_{cx}x^{(t)} + W_{ch}h^{(t-1)} + b_c) \quad (8)$$

$$h^{(t)} = \tanh(c^{(t)}) \circ o^{(t)} \quad (9)$$

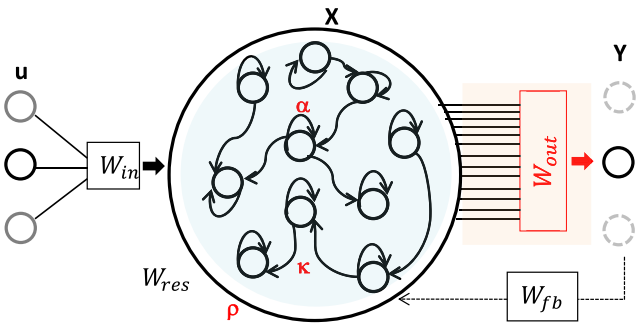

Fig. 3 Layout of an echo state network, in our experiment used with leaky neurons. The hyperparameters correspond to the ones in Table 1

Table 1 Reservoir parameters: ρ is the spectral radius, α is the leak rate, κ denotes connectivity, and λ is the regularization coefficient

Hyperparameter	Range
Reservoir size	400
κ	0.1
Input scaling	[1, 5, 9, **13**]
ρ	[0.1, 0.4, 0.7, **1.0**, 1.3]
α	[0.1, **0.3**, 0.5, 0.7, 0.9]
λ	[**0.01**, 0.1, 1, 10]

Input scaling is a factor with which the input is multiplied before being feed into the network

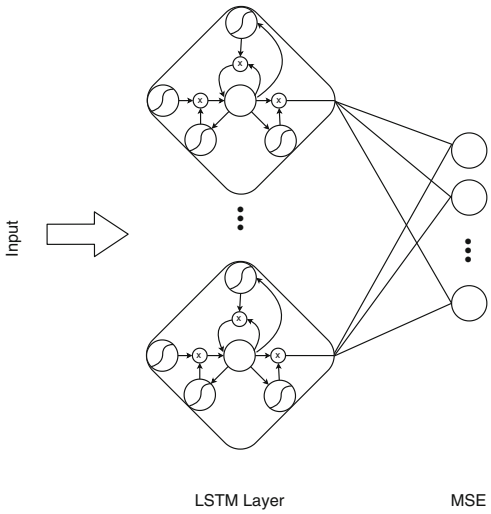

Fig. 4 The implemented LSTM architecture. We used only one recurrent layer but performed a grid search over the number of cells

where $i^{(t)}$, $f^{(t)}$, and $o^{(t)}$ denote the input, forget, and output gates respectively. $x^{(t)}$, $c^{(t)}$, and $h^{(t)}$ are the LSTM unit input, cell state, and hidden state respectively. W and b represent the weights and biases, σ is the sigmoid activation function, and $\circ$ is the element-wise multiplication.

We implemented a simple recurrent network in PyTorch[2] consisting of one LSTM layer and a 10-cell fully connected linear readout layer with no bias stacked on top. Opposed to the ESN, where the different sensors have been manually tuned [24], we now scale each channel of each sensor individually to have a standard deviation of 1. We tried 10, 20, 40, 80, and 160 recurrent cells in the hidden layer but found that only small gains in F1 score and accuracy could be achieved when increasing the number of neurons higher than 40 and, therefore, chose 80 LSTM cells for further experiments.

Although we are in a classification setting we treat the task as a regression problem and use the mean-squared error for training. This is due to the fact that we integrate gesture pauses, i.e., sequences with no gestures, in between the gestures, during which we want no neuron to be active. When using categorical cross-entropy, and therefore "softMax" as the final activation function, the network can not represent "no gesture" sequences. We use the "adam" optimizer [16] with a learning rate η set to 0.001, and train for a maximum of 100 epochs. We use 10% of our train set as a validation set and apply early stopping once the validation error starts to increase.

Evaluation Scheme and Results

The major challenge in continuous gesture recognition is the gesture spotting followed by the correct classification of the actual gesture. While the gesture spotting is problematic due to variable-length gestures, the classification is often hampered by so-called *subgestures*. Dynamic gestures, such as commands, often share the same start movements, e.g. lifting an arm or the phone. Therefore, the whole gesture sequence has to be parsed first and then mapped to the correct gesture label. Figure 5 demonstrates the problem exemplary taken from our LSTM model: while the whole sequence is a *shake*, the start is misclassified as *snap*.

In our previous study [24], we introduced an evaluation mapping scheme from actual gesture sequences to their corresponding ground truth sequences (Fig. 6). Due to the variable-length sizes of the gesture samples and the "no gesture" condition, we suggested the following: We apply a ReLu non-linearity over the network output and sum up all remaining activity at every time step. If the total activity is above a predefined threshold of 0.4 we start summing up all individual outputs over time until the total activity falls below the threshold again. The whole segment is then labeled as belonging to the class of the neuron that had the highest total activity. On the resulting segments, we run our mapping algorithm:

- Only one true positive (TP) mapping is allowed
- A wrong prediction is counted as "wrong gesture" (WG)
- A prediction and class segment that does not overlap is counted as false positive (FP)
- An actual class without a mapping is a false negative (FN)

Table 2 reports the average accuracy and F1 score of the test sets averaged over all subjects. The values in parentheses denote the standard deviation. We also provide

[2] https://pytorch.org/

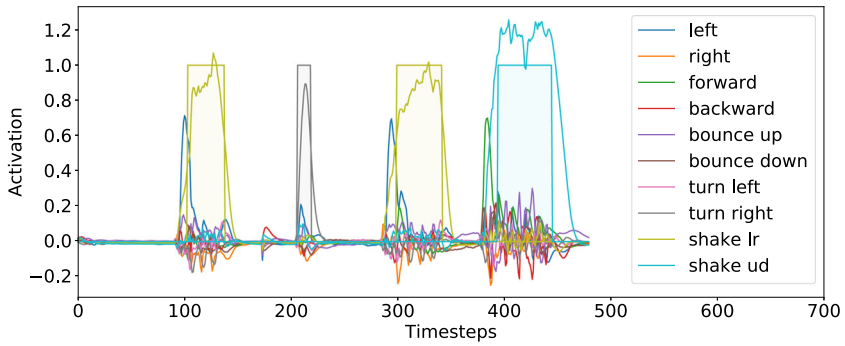

Fig. 5 Example of the output activation from the LSTM architecture. Activation for all 10 output neurons is plotted over the first 500 time steps, the ground truth is shown with transparent boxes. The subgesture problem shows very nicely for the *shake* gestures: The network commonly predicts a *snap* gesture during the early, ambiguous phase and only after a whole shake is performed the *shake* neuron turns active

the corresponding training times for comparison. While the LSTM shows superior performance for the F1 score, the difference in accuracy for both models is less apparent. The table also shows less training times for the ESN framework.

As explained before, gesture sequences in a continuous stream are easily misclassified due to *subgestures* and variances in the performance of a gesture. Therefore, we analyzed the results from each participant for both the ESN and the LSTM to explain the possible error sources. Table 3 shows the individual evaluations for the ESN and the LSTM. The different F1 scores among the subjects visible in the table indicate high *inter-subject variability*. The values in the parentheses denote the standard deviation, which is low for all subjects.

We show in Figs. 7, 8, and 9 the worst, average, and best performance from the set of our participants evaluated from our ESN (the participant names are anonymous). The misclassifications result from the average of all trials per subject, which explains why the values are not integers. Most confusion between the individual gestures is shown in Fig. 7. Noteworthy, the gestures {*snap left, snap right*} are confused with the longer gesture *shake left-right*, which supports that the *subgesture* problem affects the performance. Similarly, the *snap backward* gestures is misclassified for *shake up-down*. Further, the directions in the *shake* gestures are confused between *up-down* and *left-right*. Notably, also the "no gesture" negatively influences the performance by producing misclassification with almost all gesture classes (cf. bottom row and final column in Fig. 7). The effect of a possible prefix gesture is also shown in Fig. 8 for the *snap left* gesture, however, most of the misclassifications stem from wrong predictions of the "no gesture" class. Finally, Fig. 9 shows the highest score in the gesture performance, emphasizing again the confusion of "no gesture" with other gesture sequences.

The right column of Table 3 demonstrates the results from each participant for the LSTM. Figures 10, 9, and 12 provide insights into the misclassification from the worst to the best performance as produced by our LSTM experiments. Again the overall performance is affected by confusion of *subgestures* like *snap left* with the according *shake* gesture. The *snap backward* gesture is mostly confused with *shake up-down*. Moreover, the symmetry of the *bounce* gesture has negatively influenced the prediction results for the best performance (Fig. 12). Similarly, the average performance as shown in Fig. 11 results primar-

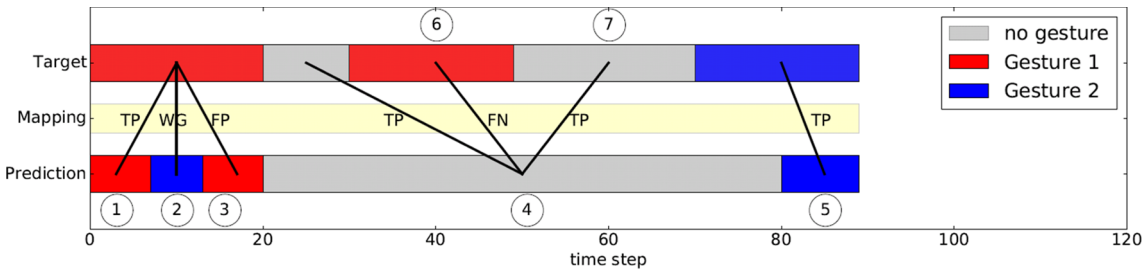

Fig. 6 Example of our proposed mapping scheme [24]. The target gesture stream consists of gesture 1, followed by a silent phase or "no gesture," another performance of gesture 1 and a subsequent pause and finally gesture 2. In case number 2, the erroneous prediction results in a "wrong gesture" followed by a false positive as we constrained gesture segments to be mapped as correct only once. The false negative is a consequence of the long "no gesture" prediction (case number 6)

Table 2 Average F1 score, accuracy on the test sets, and training times

	ESN	LSTM
F1 score	0.78 (0.09)	0.87 (0.07)
Accuracy	0.87 (0.03)	0.93 (0.04)
Train time (in sec)	2.6 (0.03)	88.9 (15.1)

Standard deviations are given in parentheses

ily from misclassifications of the *snap left* gesture. Finally, the best performance shown in Fig. 10 has the most prominent confusion between the symmetric *bounce up-down* gestures.

Discussion

We presented an experimental study on continuous gesture recognition to compare the performance between an echo state network (ESN) and Long Short-Term Memory (LSTM) network. Both networks are special architectures of recurrent neural networks and successfully applied to sequence processing. Given the inherent variances on how to perform gestures, so-called *inter-subject variability*, we were interested in the computational performance of both approaches as they are conceptually different. We used a dataset introduced in an earlier work [24] for training and testing both models. Our evaluation showed that our ESN framework achieved an accuracy comparable to the LSTM, while being faster to train (cf. Table 2). Therefore, we conclude ESNs to be a viable model for continuous gesture recognition using sensor data and, due to their fast recognition times, to be an ideal candidate for tasks that require real-time processing. The applications are manifold and range from simple body tracking to concrete rehabilitation of the limb apparatus as shown for, e.g., sEMG devices [29]. Recent approaches primarily use simple techniques like sliding windows and DTW or an SVM. However, all the techniques need special tuning of the window size or the specific kernel [27], which led to an additional subject customization step in the system [11]. In our study, we show how ESNs learn different activity patterns while being able to distinguish between gestures, subtle movements, and pauses. This highly facilitates *gesture spotting* where no extra hardware is needed [22]. Moreover, the working principles behind both approaches presented here address time-varying patterns in contrast to studies using a fixed time window [27, 29], which shadows the *inter-subject variability* problem. The nontrivial problem of learning these variances is highlighted by the study presented by Wang and Li [27]: while their system achieved 86.99% accuracy for a fixed number of gestures concatenated into one stream, the performance drops to 60% for arbitrary sequences. In contrast, our experiments achieved an average accuracy of 93% for LSTM and 87% for the ESN using randomly shuffled gestures. Learning distinct gesture patterns performed by human subjects is key to a flexible recognition system, which provides an intuitive interface without preliminary customer calibrations or fixation of gesture lengths to arbitrary values. We believe that our experiments show a new research direction in a domain that is still dominated by classic computational techniques like DTW.

Another important factor affecting the performance are *subgestures* and symmetric gestures [11]. In our experiments, the individual gestures *snap left* and *snap right* as well as the *snap forward* and *snap backward* gestures were often falsely classified with their corresponding *shake* gestures in both networks. Also, classification errors occurred between *snap up* and *snap down* with *snap forward* and *snap backward*. We hypothesize that those sequences are too short and, as we did not provide any help on the actual gesture performance, the participant might have held the phone such that the sensors detect a backward motion. The influence of misclassification caused by the issues described is more prominent for our ESN framework. The individual evaluation revealed many errors in the upper triangular part of the confusion matrix for the worst performance (cf. Fig. 7), which relates to a bad recall metric. As a result, the recall negatively impacts the F1 score for the ESN, which is 9% worse than the LSTM. However, the difference in the overall accuracy is less pronounced.

Interestingly, the performance of participant *J* changed from worst in the ESN model to best for the LSTM and vice versa for the best performance of subject *L* (cf. Table 3). When looking into our data, we observed highly distinct gesture trajectories for participant *J* for each gesture while subject *L* performed each gesture similar. The better performance of the LSTM for varying gesture sequences confirms the superior performance for the F1 score and the robustness of this network to recognize variable-length gestures. We explain the switched role for subject *L* with our mapping algorithm, which is tailored to the ESN output. We think that a modular approach as presented by Carfi et al. [5] would yield a better evaluation as a subsequent classifier is explicitly used. Finally, both models were error-prone to *subgestures* and variability of gesture sequences, especially pronounced for symmetric gestures, which supports that continuous gesture recognition is a nontrivial task. More research on these aspects in the future could be useful to expand the application area.

The main question regarding the further application of our system addresses the size of the gesture vocabulary

Table 3 Individual evaluations from our ESN and LSTM experiments with one subject as test set. Standard deviations are given in parentheses

		ESN		LSTM	
Test set	Train sets	F1 train	F1 test	F1 train	F1 test
J	L, S, Ni, Na	0.97 (0.01)	0.64 (0.04)	0.98 (0.01)	0.95 (0.02)
Ni	Na, J, L, S	0.97 (0.01)	0.81 (0.05)	0.97 (0.02)	0.83 (0.04)
S	Ni, Na, J, L	0.97 (0.01)	0.80 (0.05)	0.98 (0.02)	0.90 (0.08)
Na	J, L, S, Ni	0.98 (0.01)	0.80 (0.06)	0.98 (0.01)	0.88 (0.05)
L	S, Ni, Na, J	0.96 (0.01)	0.88 (0.05)	0.96 (0.03)	0.80 (0.04)

and the number of samples in the dataset. Although our 10 gestures used in this study is a high number compared with recent research [5, 10, 18, 29], and involving 5 participants to include gesture diversity, our experiments are evaluated only on a total of 500 sequences. We obtained high performance for the LSTM both in F1 score and accuracy, the latter comparable with our ESN framework. It would be interesting to support the expressiveness of our results for a larger dataset. Unfortunately, until today no benchmark dataset for sensor-based continuous gesture recognition exists, which precludes a reasonable and fair comparison of different computational architectures. We hypothesize that extending the gesture vocabulary for more and distinct gesture types together with a significant increase in the sample size will be challenging for the standard ESN architecture. Having our dataset publicly available, we hope that the data issue will gain attention from more researchers, yielding a more diverse dataset in the future. Variability in the gesture performance among the subjects but also for every gesture itself and the *subgesture* problem will then be more pronounced. It remains an open question whether more sophisticated ESN architectures like Deep ESN [9] or different ESN topologies [21] are key to upscale the gesture recognition tasks and comparing other deep learning approaches such as LSTM will shed further light on the applicability of models from different paradigms to real-world scenarios as for sensor interfaces or human-robot interaction.

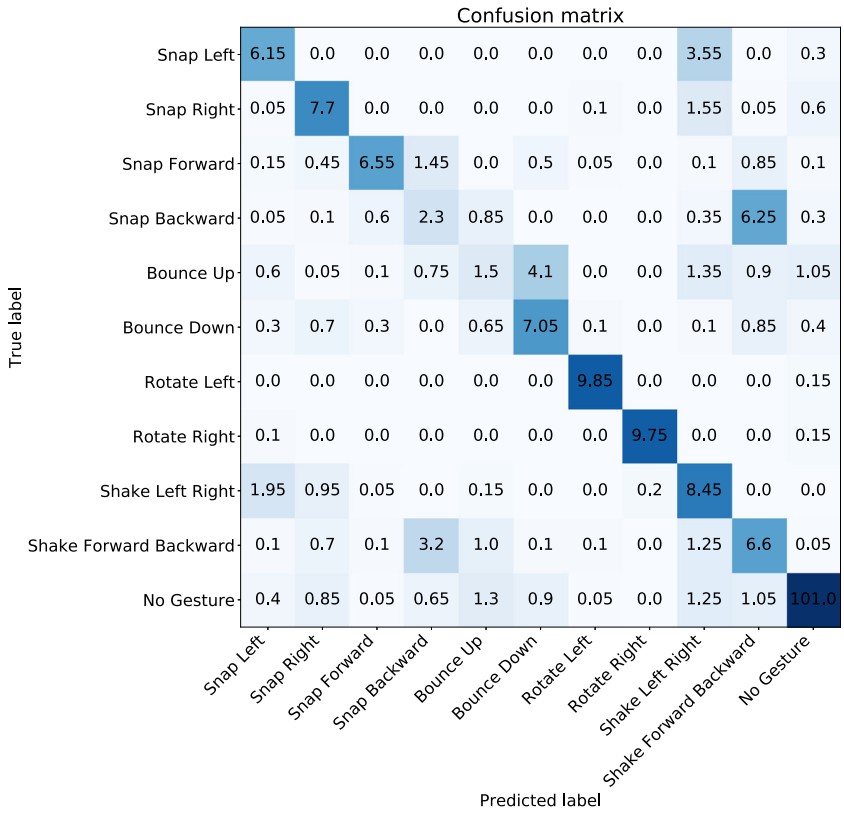

Fig. 7 Confusion matrix of the worst performance of a participant of our study resultant from our ESN. The misclassifications are mainly among short gestures predicted to be a longer but similar gesture (*subgesture*) and for symmetric gestures

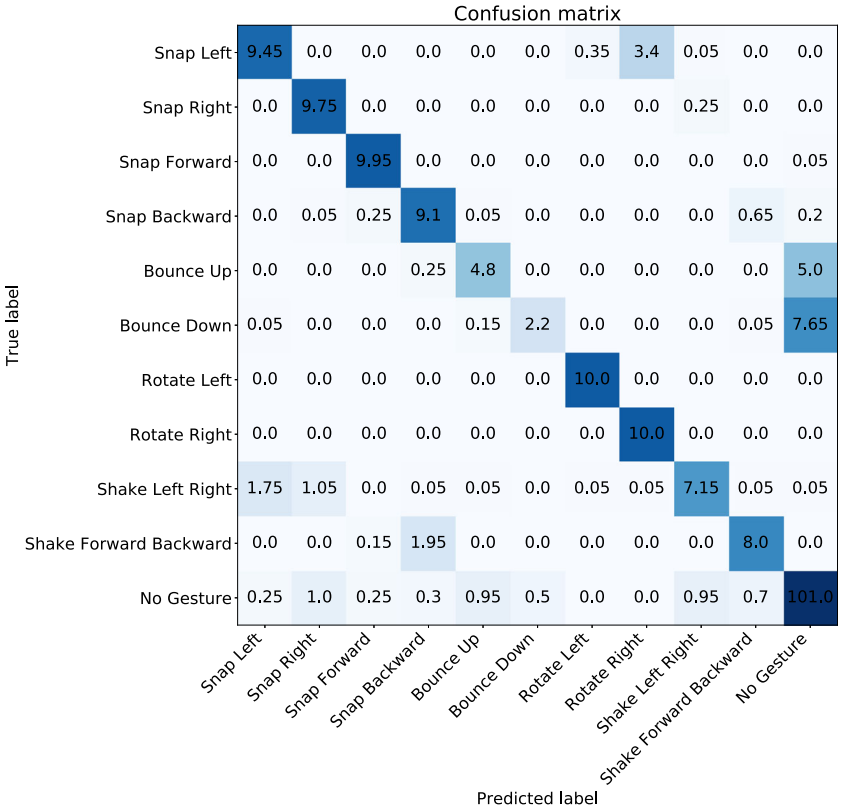

Fig. 8 Confusion matrix derived from a subject with average performance resultant from our ESN. Most confusion are between *bounce* gestures. The *subgesture* problem is less pronounced

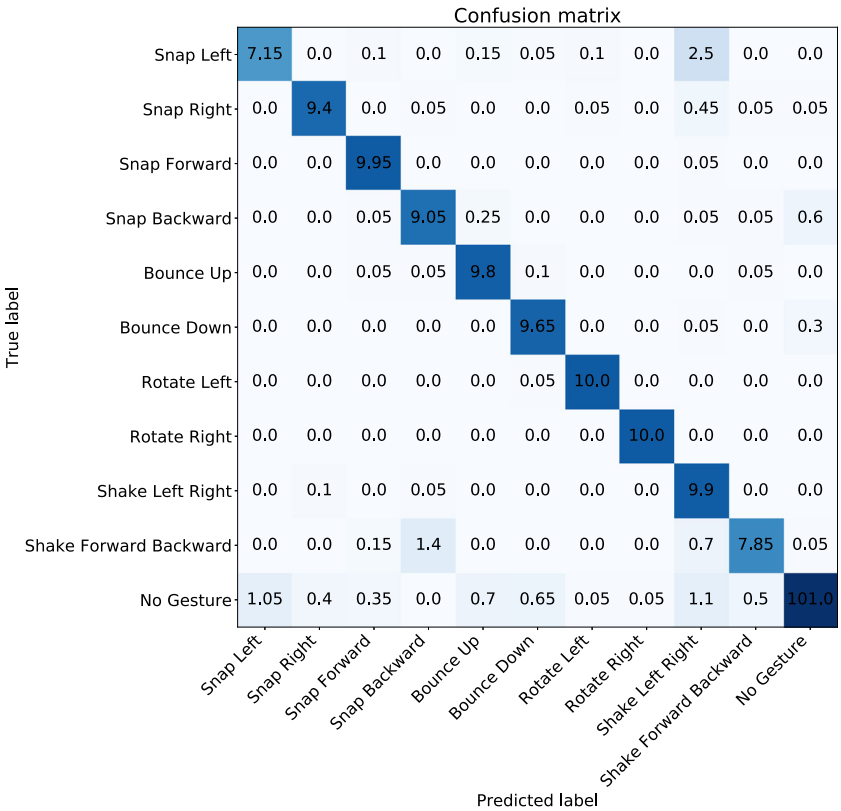

Fig. 9 Confusion matrix of the best performance among all subjects resultant from our ESN. Only a few misclassifications occur

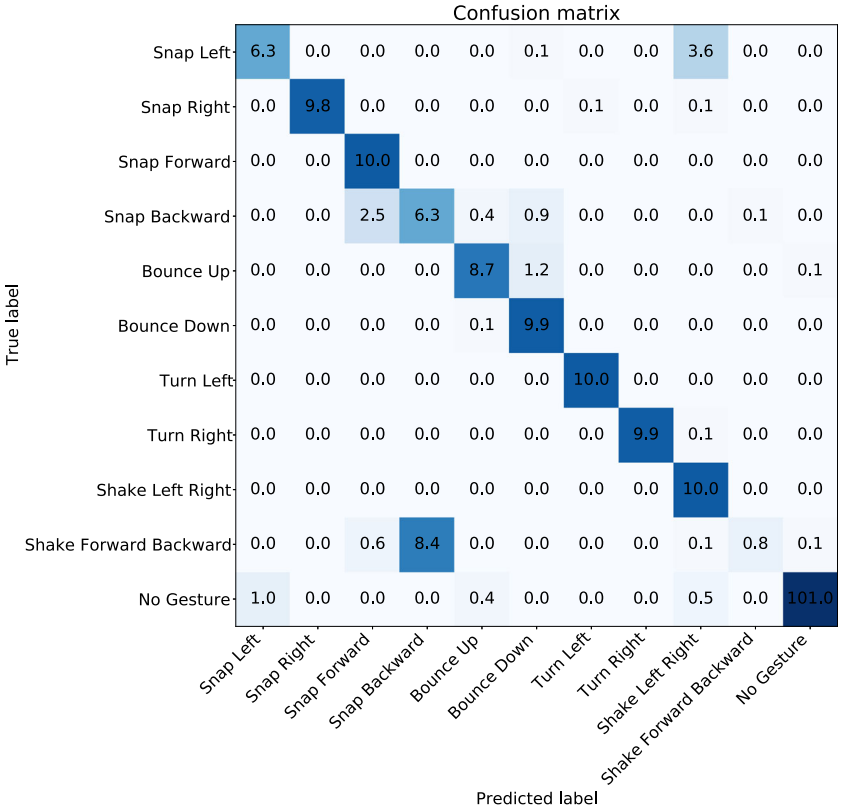

Fig. 10 Confusion matrix of the worst performance among all subjects resultant from our LSTM architecture. Most of the misclassifications stem from the *shake* gestures and their corresponding *subgestures*

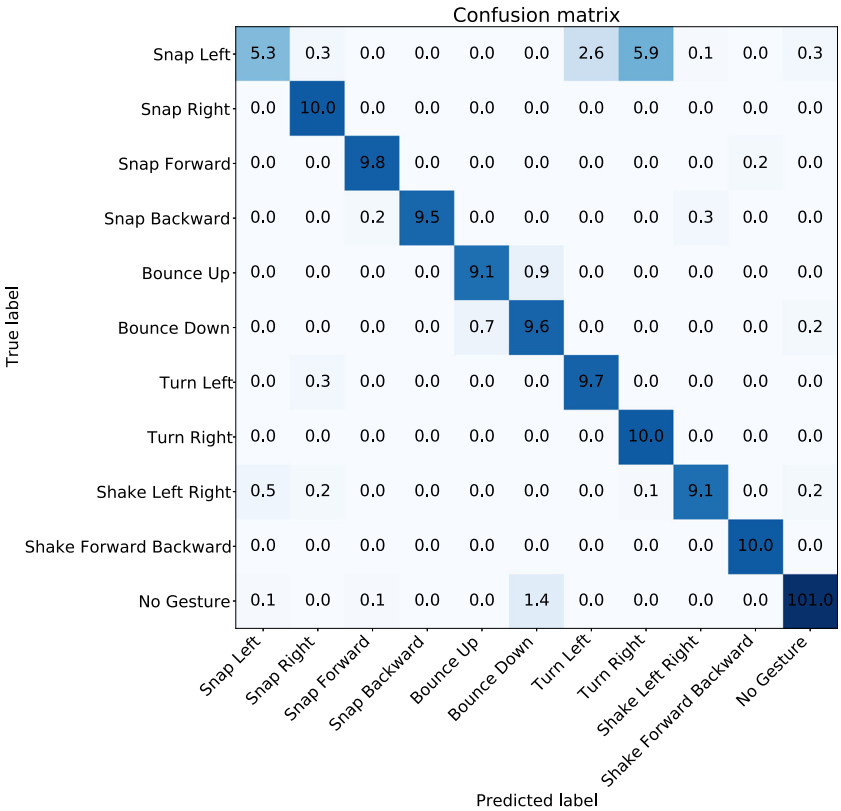

Fig. 11 Confusion matrix derived from a subject with average performance resultant from our LSTM architecture. The main source of confusion is the *left* gesture

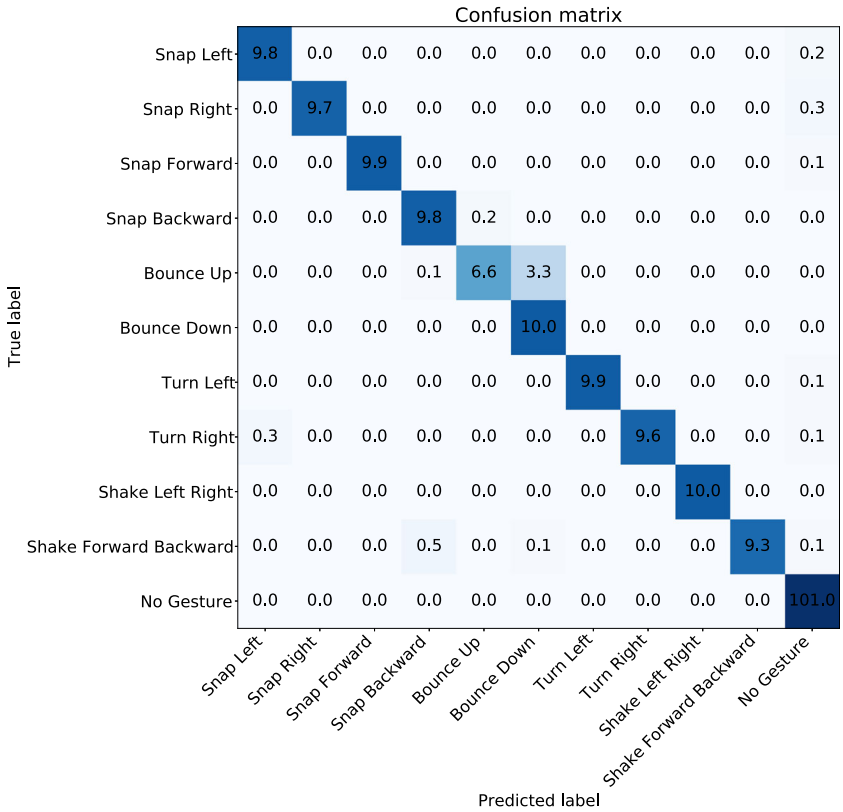

Fig. 12 Confusion matrix of the best performance among all subjects resultant from our LSTM architecture. The only significant confusion is between the *bounce* gestures

Conclusion

Continuous gesture recognition is a crucial task due to high variances of gestures in a stream and the easy confusion of inherently similar gestures. The goal of our present study was a performance comparison of a previously introduced echo state framework with a Long Short-Term Memory network, which is a state-of-the-art model for sequence processing. Our results confirm the robust processing of continuous gesture streams for both the LSTM and the ESN model, the latter showing comparable performance. As training is much faster, echo state networks are suitable computational models for experiments that require real-time processing. Our study reveal the impact of variability in the gesture performance and *subgestures* on the recognition performance for both models. We assume that these factors will be more affecting when considering a larger gesture vocabulary with more data from subjects than actually available. Until now, only little is known about the capabilities of those networks in large gesture recognition scenarios. We hypothesize that further research on echo state networks will progress to novel developments of network architectures, resulting in potential applications for many domains such as rehabilitation or human-robot interaction.

Acknowledgments The authors would like to thank the anonymous reviewers for their valuable comments on an earlier version of the manuscript.

Compliance with Ethical Standards

The authors declare that they have no conflict of interest.

Funding Information Open Access funding provided by Projekt DEAL.

References

1. Antonik P, Marsal N, Brunner D, Rontani D. Human action recognition with a large-scale brain-inspired photonic computer. Nat Mach Intell. 2019;1:530–537. https://doi.org/10.1038/s42256-019-0110-8.
2. Argyris A, Bueno J, Fischer I. Photonic machine learning implementation for signal recovery in optical communications. Scientific Reports. 2018;8:8487. https://doi.org/10.1038/s41598-018-26927-y.
3. Bengio Y, Boulanger-Lewandowski N, Pascanu R. Advances in optimizing recurrent networks. In: IEEE International conference on acoustics, speech and signal processing; 2013. p. 8624–8628.
4. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks. 1994;5(2):157–166. https://doi.org/10.1109/72.279181.
5. Carfi A, Motolese C, Bruno B, Mastrogiovanni F. Online human gesture recognition using recurrent neural networks and wearable sensors. In: 2018 27Th IEEE international symposium on robot and human interactive communication RO-MAN; 2018. p. 188–195.
6. Dasgupta S, Wörgötter F., Manoonpong P. Information dynamics based self-adaptive reservoir for delay temporal memory tasks. Evolving Systems. 2013;4(4):235–249.
7. Escalera S, Baró X., Gonzàlez J., Bautista MA, Madadi M, Reyes M, Ponce-López V., Escalante HJ, Shotton J, Guyon I. Chalearn looking at people challenge 2014: Dataset and results. Computer vision - ECCV 2014 workshops. In: Agapito L., Bronstein M. M., and Rother C., editors. Cham: Springer International Publishing; 2015. p. 459–473.
8. Gallicchio C, Micheli A. A reservoir computing approach for human gesture recognition from kinect data. In: S. Bandini, G. Cortellessa, F. Palumbo (eds.) Proceedings of the Artificial Intelligence for Ambient Assisted Living 2016 co-located with 15th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2016), Genova, Italy, November 28th, 2016, CEUR Workshop Proceedings, vol. 1803, pp. 33–42. CEUR-WS.org; 2016.
9. Gallicchio C, Micheli A, Pedrelli L. Design of deep echo state networks. Neural Netw. 2018;108:33–47. https://doi.org/10.1016/j.neunet.2018.08.002.
10. Gupta HP, Chudgar HS, Mukherjee S, Dutta T, Sharma K. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. IEEE Sensors J. 2016;16(16):6425–6432. https://doi.org/10.1109/JSEN.2016.2581023.
11. Han H, Yoon SW. Gyroscope-based continuous human hand gesture recognition for multi-modal wearable input device for human machine interaction. Sensors. 2019;19:11.
12. Hinaut X, Dominey PF. Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing. PloS one. 2013;8(2):e52946.
13. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9:1735–80.
14. Jaeger H. Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach GMD-Forschungszentrum Informationstechnik. 2002.
15. Jirak D, Barros P, Wermter S. Dynamic gesture recognition using echo state networks. In: Proceedings of the European Symposium of Artificial Neural Networks and Machine Learning, pp. 475–480; 2015.
16. Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014. arXiv:1412.6980.
17. Lee MC, Cho SB. Mobile gesture recognition using hierarchical recurrent neural network with bidirectional long short-term memory. In: Proceedings of UBICOMM; 2012. p. 138–141.
18. Li Y, Wang T, Khan A, Li L, Li C, Yang Y, Liu L. Hand gesture recognition and real-time game control based on a wearable band with 6-axis sensors. 2018.
19. Liu J, Zhong L, Wickramasuriya J, Vasudevan V. uwave: Accelerometer-based personalized gesture recognition and its applications. Pervasive and Mobile Computing. 2009;5(6):657–675. PerCom 2009.
20. Lukoševičius M., Jaeger H. Reservoir computing approaches to recurrent neural network training. Computer Science Review. 2009;3(3):127–149. https://doi.org/10.1016/j.cosrev.2009.03.005.
21. Qiao J, Li F, Han H, Li W. Growing echo-state network with multiple subreservoirs. IEEE Trans Neural Netw Learn Syst. 2017;28(2):391–404. https://doi.org/10.1109/TNNLS.2016.2514275.
22. Sosin I, Kudenko D, Shpilman A. Continuous gesture recognition from semg sensor data with recurrent neural networks and adversarial domain adaptation. In: 2018 15Th international conference on control, automation, robotics and vision (ICARCV); 2018. p. 1436–1441.
23. Tanaka G, Yamane T, Héroux JB, Nakane R, Kanazawa N, Takeda S, Numata H, Nakano D, Hirose A. Recent advances in physical reservoir computing: A review. Neural Networks. 2019;115:100–123.
24. Tietz S, Jirak D, Wermter S. A reservoir computing framework for continuous gesture recognition. In: Tetko, I. V., V. kůrková, P. Karpov, F. Theis. Artificial neural networks and machine learning – ICANN 2019: workshop and special sessions, pp. 7–18 Springer International Publishing Cham; 2019.
25. Triefenbach F, Jalalvand A, Schrauwen B, Pierre Martens J. Phoneme recognition with large hierarchical reservoirs. In: J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, A. Culotta (eds.) Advances in Neural Information Processing Systems 23, pp. 2307–2315. Curran Associates, Inc. http://papers.nips.cc/paper/4056-phoneme-recognition-with-large-hierarchical-reservoirs.pdf; 2010.
26. Verstraeten D, Schrauwen B, D'Haene M, Stroobandt D. An experimental unification of reservoir computing methods. Neural Networks. 2007;20(3):391–403. Echo State Networks and Liquid State Machines.
27. Wang Y, Ma H. Real-time continuous gesture recognition with wireless wearable imu sensors. In: 2018 IEEE 20Th international conference on e-health networking, applications and services (healthcom); 2018. p. 1–6. https://doi.org/10.1109/HealthCom.2018.8531095.
28. Wyffels F, Schrauwen B. Design of a central pattern generator using reservoir computing for learning human motion. In: 2009 Advanced technologies for enhanced quality of life, pp. 118–122; 2009.
29. Yang J, Pan J. Li, j.: semg-based continuous hand gesture recognition using gmm-hmm and threshold model. In: 2017 IEEE International conference on robotics and biomimetics (ROBIO), pp. 1509–1514; 2017.