A Reservoir Computing Framework for Continuous Gesture Recognition

Stephan Tietz[1,2](✉), Doreen Jirak[1](✉), and Stefan Wermter[1]

[1] Department of Computer Science, University of Hamburg, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
{jirak,wermter}@informatik.uni-hamburg.de

[2] Department of Computer Science, Technical University Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany
stietz@win.tu-berlin.de

Abstract. We present a novel gesture recognition system for the application of continuous gestures in mobile devices. We explain how meaningful gesture data can be extracted from the inertial measurement unit of a mobile phone and introduce a segmentation scheme to distinguish between different gesture classes. The continuous sequences are fed into an Echo State Network, which learns sequential data fast and with good performance. We evaluated our system on crucial network parameters and on our established metric to compute the number of successfully recognized gestures and the number of misclassifications. On a total of ten gesture classes, our framework achieved an average accuracy of 85%.

Keywords: Echo State Networks · Continuous gesture recognition

1 Introduction

Mobile phones rapidly advanced from a portable telephone to a device with a multitude of technological features. While most smartphones are controlled through a touchscreen or by voice, another feature that can be easily exploited is the inbuilt Inertial Measurement Unit (IMU). It enables using a phone as a direct gesture input device, similar to what has been explored by the gesture recognition community adopting the Wii Remote Controller around a decade ago.

This third, gesture based interaction channel would allow to better operate the phone when only one hand is available to the user or when the environment is too noisy to deliver speech commands. Human-robot interaction (HRI) research shows that using IMU sensor data to control a robot with gestures (e.g. showing directions in a navigation task) can be an interesting approach in contrast to vision-based approaches, which rely more on controlled environmental conditions and are thus more challenging to realize in real world applications.

Our present study introduces a framework for continuous gesture recognition exploiting the IMU sensor availability in standard mobile phones and is further

I. V. Tetko et al. (Eds.): ICANN 2019, LNCS 11731, pp. 7–18, 2019.
https://doi.org/10.1007/978-3-030-30493-5_1

motivated to foster research on our chosen model for classifying gestures: Echo State Networks (ESN) [6]. ESNs implement a specific form of recurrent neural networks (RNN) that deviate from the classic RNN training scheme. The important distinction to the usual gradient-descent methods lies in a functional separation between a neuronal 'reservoir' providing a high-dimensional processing of the input data and a read out component, in its most basic form simply a linear model.

We want to work on the time stream directly which necessitates a model that can deal with temporal data, i.e. inputs of different length. While Hidden Markov Models (HMMs) would be a more classical choice and Long Short-Term Memory Networks (LSTMs) would be more powerful, we chose ESNs as they provide a good trade off between flexibility and ease of training.

ESNs together with other models that follow a similar computational scheme are summarized under the term Reservoir Computing (RC) [12]. They have been primarily studied in the language and robotic domain including navigation [1,2,5,11] while the research on gestures is rather sparse [3,8]. This is surprising as all fields share important properties like long-range dependencies and high performance variances across subjects (e.g. the speakers' speed, gesture movement speed).

In the present paper, we address the task of continuous gesture recognition implemented within an ESN-based framework. We highlight the specific requirements for the segmentation and classification of gestures embedded in a continuous data stream recorded from movements using a smartphone. Our study shows that ESNs are a potentially interesting computational method for this task and give further directions to extend our framework for future applications in the gesture recognition domain.

2 Data Recordings and Methods

We defined a gesture protocol to standardize our data recordings and developed a 'SensorTracker App' to enable experimental reproducibility. The data and the code including the ESN are publicly available[1]. We collected IMU sensor data from a Samsung Galaxy S5 neo smartphone (Android OS). Figure 1 displays the ten gestures we used in our experiments (following [9]) recorded from a total of five subjects. The subjects were seated at a table and instructed to hold

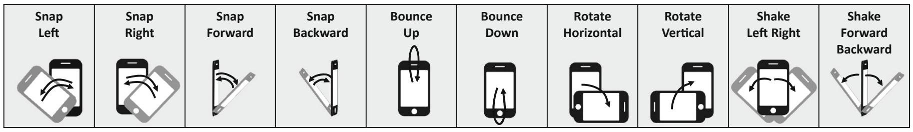

Fig. 1. Gestures performed with a smartphone used in our experiments. Arrows show the direction. Temporary positions are shown in grey.

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

the phone in their dominant hand. Each gesture performance started in the so-called 'portrait' mode (i.e. the user faces the display) with elbows at around 90°. A supervisor responsible to set the time markers (i.e. gesture start and end phase) was seated opposite to the participant but gave no instructions on how to perform the gesture specifically. Each gesture type was performed ten times with set time breaks in between to avoid fatigue and task habituation. The access to the sensor values including preprocessing steps to gain reasonable input values are described in the reference work [9]. Each time step is described by nine values representing the 3D acceleration, 3D rotation velocity (local coordinate system) and 3D orientation (world coordinate) system. Figure 2 shows an exemplary input stream for the *snap right* gesture and the markers obtained from the supervisor (blue) and our average computation (cf. Eq. 1) (green). In total, we obtained 500 gesture samples, from which we created activity-based ground truth labels:

$$target(t) = \begin{cases} 1, & 1/l * \sum_{i=t-l}^{t} \sum_{s}^{signals} |input(i,s)| \geq \theta \\ 0, & \text{else} \end{cases} \tag{1}$$

where t is the current time step, l the length of a sliding window and θ denotes a threshold. We used an input-driven ESN, so we analyzed our data before feeding it into the network regarding temporal variations and input changes, as this can have an influence on the parametrization. Figure 3 demonstrates the experimental variances across gestures and subjects. Generally, almost all gestures show only small deviations between subjects (small boxes) except outlier for *snap right* and *snap forward* and performances range in the interval of 10 to 40 time steps (≈0.3 to 1.3 s). In contrast, the *shaking updown* and *shaking left-right* gestures took around 50–70 time steps (≈1.6 to 2.3 s) and showed highest variability regarding the subjects performances and between the two gestures.

Also, we consider gestures that differ in the required rotation (rot) or acceleration (acc). We call the performance strength the 'energy':

$$Energy(t) = E_{acc}(t) + E_{rot}(t) \tag{2}$$

$$E_{acc}(t) = \sqrt{acc_x(t)^2 + acc_y(t)^2 + acc_z(t)^2} \tag{3}$$

$$E_{rot}(t) = \sqrt{rot_x(t)^2 + rot_y(t)^2 + rot_z(t)^2} \tag{4}$$

The gesture energy is summed over time t and the value is normalized by dividing its sample length to avoid a bias towards longer gesture samples. Notably, the absolute rotation value is constantly smaller than the acceleration due to the different scales (rad/s vs. m/s^2). This fact has to be taken into account for the input scaling before feeding the data into the reservoir.

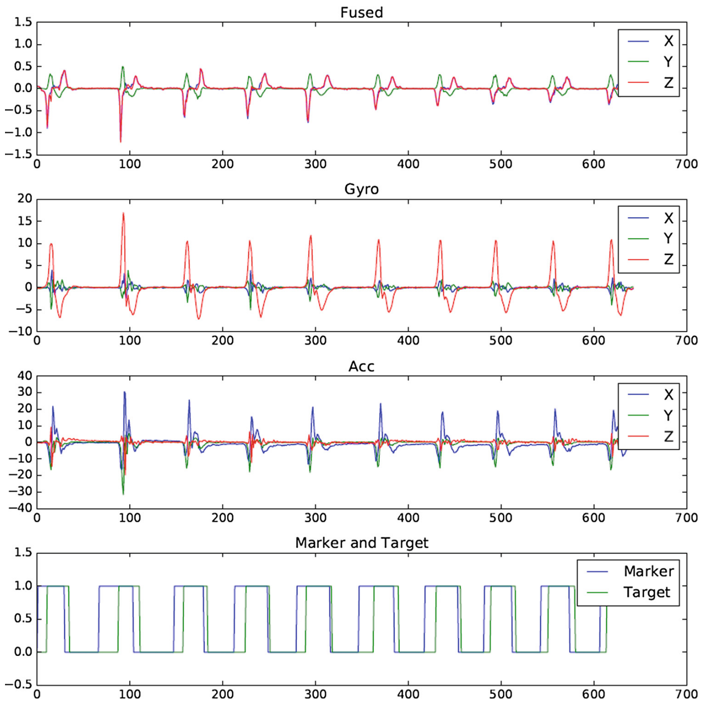

Fig. 2. Sensor data for the *snap right* gesture. (Color figure online)

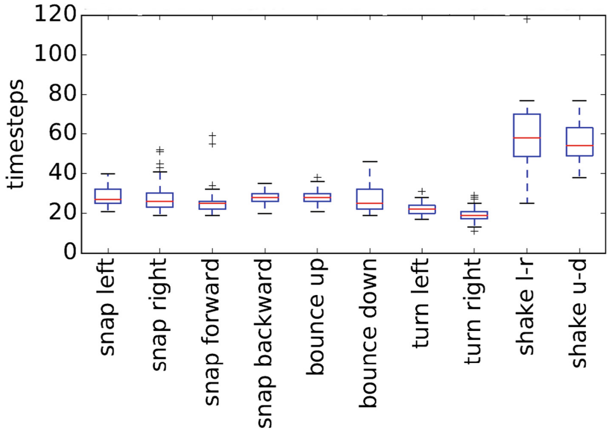

Fig. 3. Gesture variances within gestures and across the different gesture types. Black crosses denote outliers. (Color figure online)

2.1 Echo State Networks

Echo State Networks (ESN) [6] have proven successful in prediction benchmarks including chaotic time series. Here, we follow the standard notation for leaky-integrator ESN (LI-ESN) [7] and describe our scheme to search for optimized network parameters.

An ESN is functionally separated into a reservoir of recurrently connected neurons, whose activation, or simply states, x are updated as:

$$\tilde{x}(t+1) = f(u(t+1)W^{in} + x(t)W^{res} + y(t)W^{fb} + \nu(t)) \quad (5)$$

$$x(t+1) = (1-\alpha)x(t) + \alpha\tilde{x}(t+1) \quad (6)$$

where f is the activation function (here $tanh$) and u is the input. 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. W^{res} is initialized fully connected with gaussian weights and then rescaled 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.

The neural activation are collected into a state matrix X and then fitted using a linear model. Given the output Y (supervised learning):

$$Y = g(W^{out}X) \quad (7)$$

with linear output function g, standard ESN training contracts to a regression on the output weights W^{out}. Using common matrix notation for regression it can be computed:

$$W^{out} = YX^T(XX^T + \lambda\mathbf{I})^{-1} \quad (8)$$

where λ is a penalty term applied in ridge regression and $\mathbf{I}$ is the identity matrix.

Our input vectors contain nine sensor values and the output layer has ten neurons representing the gesture classes. The fact that temporal patterns excite the reservoir differently can be represented in the readout [10]. Figure 4 shows three possible shapes for the target signal. First, a rectangular shape whenever a signal is detected, which we obtain when applying our threshold scheme explained below. Second, a Gaussian shape providing a model for the low activity in the beginning of every gesture (start and end) and greatest activity during the gesture performance (stroke). The last option creates a time window that focuses only on the last time steps assuming that most information is gathered after sufficient reservoir excitement. In our experiments, we obtained the best results when using the first option (Fig. 4a).

2.2 Training and Parameter Validation

The data is split into a $n-1$ persons training set, where it is further split into segments containing only one target sample, shuffled randomly and then concatenated to four different training sets. The remaining person is used as the test set. We performed a grid search to obtain a reasonable network parametrization.

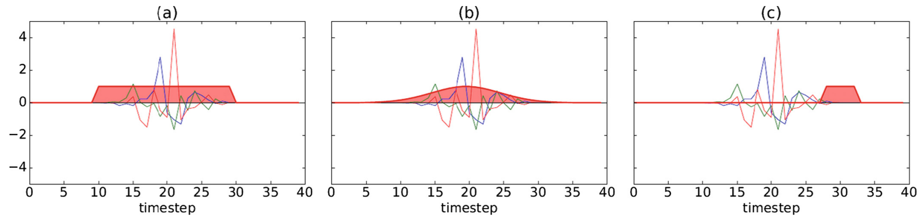

Fig. 4. Sketch of three different shapes for the target signal whenever activity occurs. (a) A rectangular shape (b) A Gaussian shape (c) A time window considering only the final activity.

For each combination of model parameters, an ESN is evaluated via leave-one-out cross validation. As this process is an exhaustive search over the parameter space, Table 1 shows the final collection of values, where we show in bold the chosen values giving the lowest validation error and which are used for the gesture experiments.

Table 1. Reservoir Parameters: Here ρ stands for spectral radius, α for the leak rate, κ for input connectivity and λ for the regularization coefficient from ridge regression.

Parameter	Domain
Reservoir size	400
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
κ	0.1
λ	**0.01**, 0.1, 1, 10

In summary, we observe from our validations on network parameters, that obeying the reciprocal relationship between connectivity and input scaling as well as keeping the processing power in the reservoir by a low leakage rate α and a (relatively) high spectral radius ρ is beneficial for our task.

3 Classification Issues in a Continuous Data Stream

In many gesture recognition tasks the network is presented with a short segment that only contains one isolated sample of a gesture with predetermined start-stroke-end scheme.

In strong contrast to this approach we train our network on a continuous time stream containing multiple, different gestures. The network does not only need to learn which gesture it sees but also when each gesture starts and ends.

This is a much more realistic, but also more challenging, setup than the classical classification task.

We now explain how we detect and distinguish between active segments and nonactive gesture segments and their corresponding labels. This is an important process to ensure proper network evaluation.

3.1 The 'No-Gesture' Class

As we work on a continuous time stream, the network outputs label information at every time step, as opposed to just a single label after the gesture has been presented. Therefore we need a way to deal with segments in between gestures that do not contain any information.

The simplest approach would be to use a constant cutoff and only label a time step if any neurons activity is above the threshold. A more flexible, neural way would be to add another neuron to the output layer that functions as a 'No-Gesture' class. The target for this neuron is to be active whenever the smart phone is resting and to be inactive while a gesture is being performed, i.e. it is one whenever all other targets are zero.

However, our experiments reveal that the activation of this signal can lead to unstable behavior, consequently producing misclassifications, as demonstrated in Fig. 5. While for some active segments the threshold signal decreases, e.g. at time step 4800, and a *turn left* gesture is classified, at around time steps 5050–5150 the threshold oscillates, distracting the classifier.

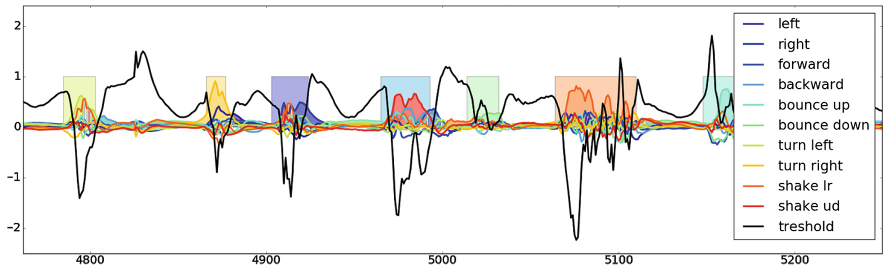

Fig. 5. Example of a threshold (black line) learnt to discriminate between an active and a non-active gesture segment. The colored areas show the actual gesture hypotheses. Its activity increases when no gestures are classified and decreases otherwise. The signal also tends to fluctuate with some negative impact on the classification performance of our system, although overall, the approach is robust. (Color figure online)

We run an exemplary experiment on a random subset of our dataset comparing the two approaches, the additional no gesture output-neuron versus a fixed cutoff value. We obtained an average F1-score of 0.874 for the neural approach and an average F1-score of 0.949 for a fixed threshold value of 0.5. Consequently we chose the constant threshold method for further evaluations.

3.2 Gesture Segmentation Process

In a continuous gesture stream, the determination of start and end as well as the distinction between a gesture performed and 'No-Gesture' are specific challenges of gesture recognition systems. As we are using supervised learning, a first approach to tackle this issue would be to segment gestures when the predicted label changes. This is problematic due to the so-called subgesture problem. A subgesture is when a specific gesture can either have a standalone character with its own meaning or be an integral part of another gesture. Figure 6 demonstrates how subgestures can influence the classification. We differentiate between the 'ArgMax' approach, which at every time step simply hypothesizes a gesture based on the neuron with highest activity and the actual desired segmentation. At the beginning of this example the 'ArgMax' approach classifies a *snap forward* gestures but in effect the gesture is a *shake up-down*. The former gesture is a subgesture of the latter, and as such every time a *shake* gesture is performed it is highly likely that misclassifications occur which reduces the overall gesture system performance.

To avoid this issue we start summing up the activity of every output neuron over time, as soon as it hits a certain threshold. When activity of all neurons falls below the threshold we label the gesture based on the neuron with highest total activation since the threshold has been exceeded.

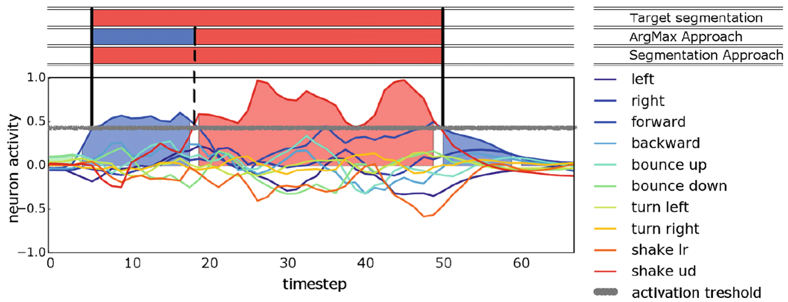

Fig. 6. Gesture segmentation showing the effect when gestures have the same subgesture. Here, the actual gesture is *shake up-down* but the first movements are detected as *snap forward*.

With the resultant segmentation we proceed to identify a mapping between the predicted outcome of the classification and the true labels. This mapping is important to provide a meaningful evaluation scheme on correctly but also incorrectly classified gestures. Notably, the prediction and target segments do not need to have matching segmentation borders. Further they do not need to have the same number of segments. Our rules for the matching between the (segmented) gesture prediction and the actual correct gesture are as follows:

- A prediction and a target segment are mapped together when they are overlapping in at least one time step. No care is given whether the length of the predicted segment matches the original signal length.
- Each predicted gesture segment is mapped exactly once.
- There is only one true positive (TP) which counts as the target gesture segments; otherwise, the segment is counted as false positives (FP) or wrong gestures (WG).
- If a predicted segment does not overlap with a target class signal, it will be mapped and labeled as WG. It will be mapped to 'No-Gesture' and scored as FP if neither the original class nor any other class can be mapped.
- If a target segment remains without any mapping, it will be marked as a *'No-Gesture'* segment and labeled as false negative (FN).
- For *'No-Gesture'* segments an unlimited amount of matches is allowed.

The set of rules allows to label all segments in a sequence while preserving the temporal dependencies. An example of the rule application is demonstrated in Fig. 7. Our approach has a disadvantage when considering the definition of true positives (TP), i.e. when the actual gesture was correctly classified - it assumes a one-to-one correspondence from the target signal to the classifier signal. However, a target signal can be recognized a second time by another prediction signal, which will result in a double entry of a target signal in its corresponding row of the confusion matrix. Is this the case, the row sums define an upper bound for the positive entries per class but do not tell the absolute value anymore. A solution would be to normalize over the row sums, however, we prefer to evaluate the number of misclassifications rather than argue over normalized values.

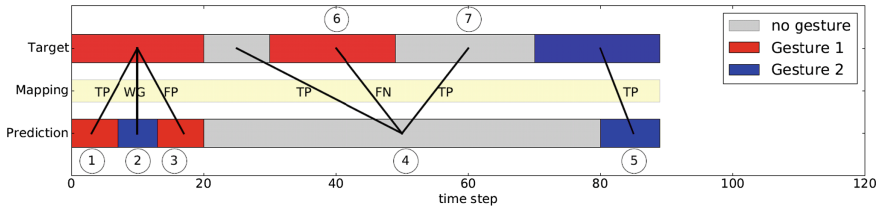

Fig. 7. Example of the mapping between predicted and actual labels when applying our rules.

4 Evaluation

We evaluated our system using the F1-score obtained from the segmentation and mapping procedure. Additionally, we evaluated our system on an individual basis to show how *inter-subject variability* across subjects can influence the final classification. Finally, we provide an overall evaluation of our system. The focus of this paper is not on achieving the highest possible performance but rather on showing a novel evaluation approach without the need for segmenting the signal before classification. We therefore refrain from reporting comparison scores.

Table 2 summarizes the results from a leave-one-out test and evaluation on each participant (names are anonymised). The varying F1-scores underpin the performance variations introduced in our experiments, however, the evaluation on gesture misclassifications gives us more valuable information. Therefore, we also provide a confusion matrix for an average subject (Ni) in Fig. 8.

In accordance with the previously mentioned subgesture problem misclassifications mostly originate from confusions of the *shake up-down* ('shake ud') gesture with the forward and backward gesture and respectively of the *shake left-right* ('shake lr') gesture with a *left* or *right* gesture.

Table 2. Subject evaluation: network is trained in a hold one out fashion on data of 4 people, then evaluated on a fifth, unseen person. Scores are reported for every fold. Train error is calculated as $1 - \mathit{F1\ score}$

Test set	Train	Train error	F1 test score
J	{L, S, Ni, Na}	0.088	0.624
Ni	{Na, J, L, S}	0.096	0.791
S	{Ni, Na, J, L}	0.076	0.809
Na	{J, L, S, Ni}	0.074	0.831
L	{S, Ni, Na, J}	0.093	0.926

The overall classification ability of our system is reported in Table 3. As described above gesture segments of four people are shuffled and mixed together into four different time streams. Validation score is calculated by leave one out cross validation on these four streams. Test score is calculated on the remaining, unseen fifth person. All results are averaged over five repetitions to account for stochastic effects during initialization and training.

Table 3. F1 scores and accuracy for final experiment. Reported values are averaged over five repetitions, standard deviation shown in brackets.

Validation		Test	
F1 score	Accuracy	F1 score	Accuracy
0.923 (0.00)	0.932 (0.00)	0.734 (0.03)	0.846 (0.02)

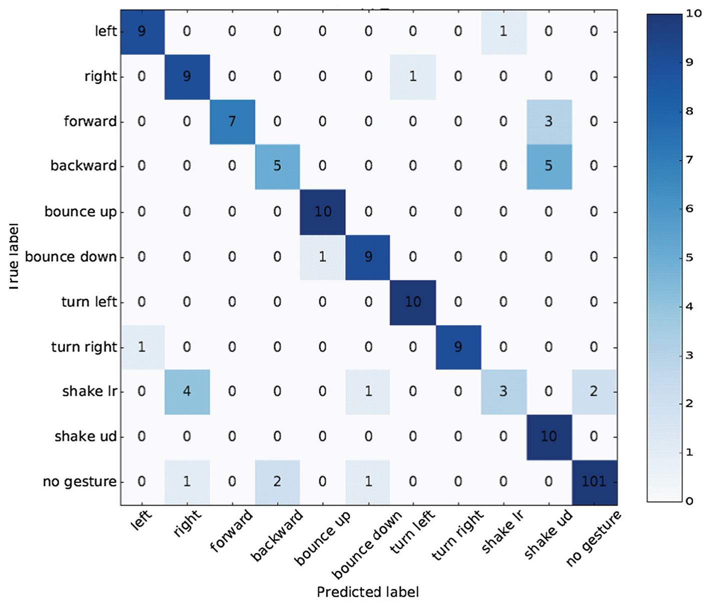

Fig. 8. Confusion matrix showing an average performance among the five subjects.

5 Discussion and Future Work

We have demonstrated a successful ESN-based framework for the task of continuous gesture recognition. We introduced a scheme for the meaningful segmentation of gestures in a stream of sensor values and their mapping to labels to establish a reasonable evaluation scheme. This way, we laid the foundation for future IMU data collections. We presented the parametrization of an ESN for optimal classification and evaluated our system on individual subjects. Although we highlighted the *inter-subject variability* as an influential factor for classification, our system achieved overall an accuracy of 85% with notably few standard deviation.

From our experiments, we propose some ideas for future applications which would improve our system. First, we are aware that our dataset is rather small compared to common gesture benchmark datasets used in machine learning. Although IMU data is used for different applications, no other, probably bigger, dataset is publicly available. Therefore, we provide our IMU dataset along with our implementation to encourage further explorations on the data including our segmentation scheme as well as analysis on the ESN model. Regarding the ESN we think of an additional online learning phase for newly introduced people to increase the system adaptability. We also recommend revising our segmentation process to be independent on the zero activity condition applied so far. For future application, the *No-Gesture* condition should be removed by a mechanism which can better distinguish between a meaningful gesture and some random movements. Finally, the emergence of deep reservoir architectures [4] offer opportunities to investigate the timescale properties of gestures, which would probably help solving the subgesture problem as we highlighted for some of our gestures, and to analyse those networks regarding their capability to cope with *inter-subject variability*, still a dominant factor influencing the performance in gesture recognition systems.

References

1. Antonelo, E.A., Schrauwen, B.: On learning navigation behaviors for small mobile robots with reservoir computing architectures. IEEE Trans. Neural Netw. Learn. Syst. **26**(4), 763–780 (2015). https://doi.org/10.1109/TNNLS.2014.2323247
2. Dasgupta, S., Wörgötter, F., Manoonpong, P.: Information dynamics based self-adaptive reservoir for delay temporal memory tasks. Evolving Syst. **4**(4), 235–249 (2013). https://doi.org/10.1007/s12530-013-9080-y
3. Gallicchio, C., Micheli, A.: A reservoir computing approach for human gesture recognition from kinect data. In: Proceedings of the Workshop Artificial Intelligence for Ambient Assisted Living (AI*AAL), vol. 1803, pp. 33–42 (2017)
4. Gallicchio, C., Micheli, A., Pedrelli, L.: Design of deep echo state networks. Neural Netw. **108**, 33–47 (2018). https://doi.org/10.1016/j.neunet.2018.08.002
5. Hinaut, X., Petit, M., Pointeau, G., Dominey, P.F.: Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks. Front. Neurorobot. **8**, 16 (2014). https://doi.org/10.3389/fnbot.2014.00016
6. Jaeger, H.: Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach. GMD-Forschungszentrum Informationstechnik (2002)
7. Jaeger, H., Lukoševičius, M., Popovici, D., Siewert, U.: Optimization and applications of echo state networks with leaky- integrator neurons. Neural Netw. **20**(3), 335–352 (2007). Echo State Networks and Liquid State Machines: https://doi.org/10.1016/j.neunet.2007.04.016
8. 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)
9. Lee, M.C., Cho, S.B.: Mobile gesture recognition using hierarchical recurrent neural network with bidirectional long short-term memory. In: Proceedings of UBICOMM, vol. 6, pp. 138–141 (2012)
10. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Comput. Sci. Rev. **3**(3), 127–149 (2009). https://doi.org/10.1016/j.cosrev.2009.03.005
11. Tong, M.H., Bickett, A.D., Christiansen, E.M., Cottrell, G.W.: Learning grammatical structure with echo state networks. Neural Netw. **20**(3), 424–432 (2007). Echo State Networks and Liquid State Machines: https://doi.org/10.1016/j.neunet.2007.04.013
12. Verstraeten, D., Schrauwen, B., D’Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. Neural Netw. **20**(3), 391–403 (2007). Echo State Networks and Liquid State Machines: https://doi.org/10.1016/j.neunet.2007.04.003