

Kent Academic Repository

Full text document (pdf)

Citation for published version

Koch, Philipp and Phan, Huy and Maass, Marco and Katzberg, Fabrice and Mazur, Radoslaw and Mertins, Alfred (2018) Recurrent Neural Networks with Weighting Loss for Early Prediction of Hand Movements. In: 26th European Signal Processing Conference (EUSIPCO 2018). , Rome, Italy pp. 1152-1156.

DOI

<https://doi.org/10.23919/EUSIPCO.2018.8553483>

Link to record in KAR

<https://kar.kent.ac.uk/72665/>

Document Version

Author's Accepted Manuscript

Copyright & reuse

Content in the Kent Academic Repository is made available for research purposes. Unless otherwise stated all content is protected by copyright and in the absence of an open licence (eg Creative Commons), permissions for further reuse of content should be sought from the publisher, author or other copyright holder.

Versions of research

The version in the Kent Academic Repository may differ from the final published version.

Users are advised to check <http://kar.kent.ac.uk> for the status of the paper. **Users should always cite the published version of record.**

Enquiries

For any further enquiries regarding the licence status of this document, please contact:

researchsupport@kent.ac.uk

If you believe this document infringes copyright then please contact the KAR admin team with the take-down information provided at <http://kar.kent.ac.uk/contact.html>

Recurrent Neural Networks with Weighting Loss for Early Prediction of Hand Movements

Philipp Koch*, Huy Phan†, Marco Maass*, Fabrice Katzberg*, Radoslaw Mazur*, and Alfred Mertins*

*Institute for Signal Processing, University of Lübeck, Germany

†Institute of Biomedical Engineering, University of Oxford, Oxford, UK

{koch, maass, katzberg, mazur, mertins}@isip.uni-luebeck.de, huy.phan@eng.ox.ac.uk

Abstract—We propose in this work an approach for early prediction of hand movements using recurrent neural networks (RNNs) and a novel weighting loss. The proposed loss function leverages the outputs of an RNN at different time steps and weights their contributions to the final loss linearly over time steps. Additionally, a sample weighting scheme also constitutes a part of the weighting loss to deal with the scarcity of the samples where a change of hand movements takes place. The experiments conducted with the Ninapro database reveal that our proposed approach not only improves the performance in the early prediction task but also obtains state of the art results in classification of hand movements. These results are especially promising for the amputees.

Index Terms—hand movement classification, hand movement prediction, electromyography, early prediction, RNN

I. INTRODUCTION

Nowadays, for many fields the determination of hand movements is useful like virtual reality. Particularly, the determination of hand movements becomes very important when it comes to the control of upper limb prostheses. In order to regain the functionalities of the hand, a variety of different hand movements has to be reliably recognized. For this application, surface electromyogram (sEMG) data are often the best modality [1]–[3].

In control systems of upper limb prostheses it is crucial to achieve two goals: (1) being reliable in hand movement determination and (2) minimizing the time delay between executing the hand movement and the system response. However, in general, the methods for determining hand movements given sEMG signals apply or adopt the standard classification scheme that introduces an unavoidable time delay [4]–[7]. Firstly, around the current time point a window of sEMG data is required for which a feature representation is calculated. This causes a time delay of half the window length, not to mention the time needed for feature extraction. Secondly, as the classification of the corresponding hand movement using the feature representation takes place, the time required for performing the classification also contributes to the time delay. Regarding the delay caused by the window length, one faces a trade-off problem. Window-wise classification produces better performance when using longer windows at the cost of a longer delay [8]. In general both long delays and unreliable classification results severely worsen the usability.

In an attempt to tackle the trade-off problem, we reformulated in recent works [9], [10] the classification problem

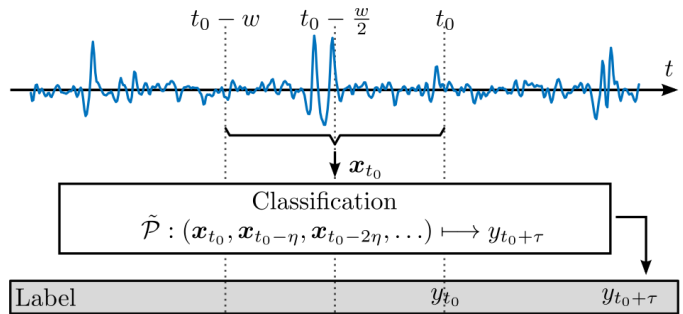


Fig. 1. Illustration of early prediction of hand movements. A label $y_{t_0 + \tau}$ is assigned to the current window represented by x_{t_0} with respect to representations of previous windows $x_{t_0 - \eta}, x_{t_0 - 2\eta}, \dots$. In our formulation of the classification task is $\tau = 0$ in contrast to the common formulation where $\tau = -\frac{w}{2}$, causing a delay of $\frac{w}{2}$. For $\tau > 0$ we name it early prediction and if then $y_{t_0 + \tau} \neq y_{t_0}$ it is called a true prediction case.

to avoid such delays. Motivated by the sequential nature of sEMG data, we proposed the early prediction of future hand movements and incorporated information of previous windows into the hand movement recognition for more reliability (see Fig. 1). By determining a future hand movement it is possible to compensate for the delay in the classification pipeline.

For predictors of future hand movements the most difficult case is seemingly the true prediction case. By that we mean the case in which the label corresponding to the end of the current window is different from the future label that should be predicted (in Fig. 1: $y_{t_0 + \tau} \neq y_{t_0}$). The accuracy in these cases is significantly lower than for the classification case. Presumably, the main reason for this is the low number of training examples for the true prediction case. In databases like the ones of the Ninapro project [11] the hand movements are quite long and there are not many transitions between hand movements. Consequently, for a vast majority of training (and test) examples the label that should be predicted is the same as the one assigned to the window. In this work, we propose an RNN-based approach for early prediction of hand movements and investigate different strategies to deal with the low number of true prediction examples. More specifically, during training the RNN, we introduce a weighting scheme to the loss function in order to make examples corresponding to the true prediction case more important than others. Furthermore, we also perform oversampling to increase the number of true prediction examples in training data. To improve the performance of the

RNN for classification and early prediction, we calculate the loss function over all time steps presented to the system. The influence of the time steps is increased in a linear way in order to make the final output the most important one.

The results on the Ninapro database [11] reveal that the proposed modifications of the training procedure help to improve both classification and early prediction of hand movements.

II. EARLY PREDICTION OF FUTURE HAND MOVEMENTS

Let $\mathbf{x}_{t_0} \in \mathbb{R}^D$ denote the D -dimensional feature representation of the sEMG window lasting from $t_0 - w$ to t_0 with t being the current time point and w being the window length. Furthermore, let $y_{t_0} \in \mathbb{L}$ where $\mathbb{L} = \{1, 2, \dots, C\}$ denotes the corresponding label specifying one of the C possible hand movements. When future hand movements should be determined, the prediction offset τ is introduced. The determination of the future movement with respect to the information of previous windows reads

$$\tilde{\mathcal{P}} : (\mathbf{x}_{t_0} \in \mathbb{R}^D, \mathbf{x}_{t_0-\eta}, \mathbf{x}_{t_0-2\eta}, \dots) \mapsto y_{t_0+\tau} \in \mathbb{L} \quad (1)$$

where η denotes the hop between consecutive windows.

If the information of previous windows $\mathbf{x}_{t_0-\eta}, \mathbf{x}_{t_0-2\eta},$ and so on is ignored and $\tau = -\frac{w}{2}$ then the formulation in (1) becomes the standard mapping used for hand movement classification.

III. CLASSIFIER AND PREDICTOR

The mapping in (1) can be modelled by different types of classifiers. However, RNNs are natural candidates due to their capability in sequential modelling. The general principle of RNNs is the processing of sequences by producing an output for each input and storing information in a state. Given a new input the old state and the input are used to calculate the output and the new state. Therefore, in an RNN cell several different weights have to be learned during a training phase.

In our system the input is iteratively fed into an RNN cell and the output of the cell is then passed to a fully-connected layer followed by a softmax layer for classification.

To allow for a comparison with previous works, we employed gated recurrent unit (GRU) [12] based RNN architecture as in [10] and use the sequence-to-label approach. That is, the accuracy of the system is determined by considering the result of the final input/output only.

The network is trained to minimize a loss function. We make use of the cross-entropy and introduce a scheme to weight the contribution of the loss at different time steps of the RNN. The most important part of this loss function is the cross-entropy E . For one training example, it is calculated as

$$E(\Theta|\mathbf{X}, \mathbf{y}_t) = -\mathbf{y}_t \log(\hat{\mathbf{y}}_t(\Theta|\mathbf{X})). \quad (2)$$

Here, \mathbf{y}_t is a one-hot encoded label vector representing the ground truth label corresponding to the last time step t . $\hat{\mathbf{y}}_t(\Theta|\mathbf{X})$ denotes the prediction of the network for the final time step given the sequence of inputs \mathbf{X} and the set of trainable parameters Θ .

Instead of calculating the cross-entropy with respect to the last time step T exclusively, the contribution of the network output at a time step t , where $1 \leq t \leq T$, into the final loss E_w is weighted by weighting coefficient $\alpha(t)$. Let $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T)$ be the sequence of one-hot encoded label vectors corresponding to a sequence of feature vectors \mathbf{X} , then, the weighted cross-entropy E_w is given by

$$E_w(\Theta|\mathbf{X}, \mathbf{Y}) = -\sum_{t=1}^T \alpha(t) \mathbf{y}_t \log(\hat{\mathbf{y}}_t(\Theta|\mathbf{X}_{1:t})) \quad (3)$$

with $\mathbf{X}_{1:t}$ denoting a part of the sequence of feature vectors upto time step t . For the experiments in this work, α is chosen to be a linear weighting in the form

$$\alpha(t) = \frac{2t}{T(T+1)}. \quad (4)$$

This weighting ensures that early time steps that are likely influenced by the initialisation of the cell state contribute less to the loss function than later time steps that should lead to more reliable predictions. The chosen linear weighting can easily be replaced by other weightings that meet the condition

$$\sum_{t=1}^T \alpha(t) = 1 \quad (5)$$

to prevent that a varying number of time steps T changes the proportion of the elements contributing to the loss function.

For further regularisation, we used ℓ_2 -norm regularisation as well as dropout on the output of the GRU cell [13]. During training, min-batches were used. For optimization of the whole network, the Adam optimizer was used [14].

IV. TRAINING ADAPTATION TO IMPROVE EARLY PREDICTION PERFORMANCE

A challenge regarding the early prediction problem is the low number of training examples for the true prediction case. Given a windowed signal covering multiple consecutive hand movements, in most cases, the label corresponding to a given window y_t is the same as the label $y_{t+\tau}$ with $\tau > 0$ that should be predicted. Thus, it is basically a classification. The early prediction case where $y_t \neq y_{t+\tau}$ is very rare. Since the true prediction case is highly under-represented in training data, the classifier will easily ignore this case in order to achieve the best possible overall accuracy. Therefore we used two strategies to improve the performance regarding these cases.

A. Oversampling

One approach to deal with the under-representation of the true prediction case is the generation of more samples for true prediction using oversampling. Let t_c denote a point in time of a label change. Consequently, all windows ending in $(t_c - \tau, t_c]$ result in a true prediction example. Usually, training examples are generated by cropping overlapping windows out of the signal. By increasing the degree of overlapping in the region of $(t_c - \tau, t_c]$ more training samples for the true prediction case are generated.

This partial oversampling remedies the unbalance between the true prediction samples and others in the training data. With this approach, it is possible to harvest more training data for the under-represented case without reusing existing training samples. However, the oversampled training samples are more correlated due to the larger overlap of the windows.

B. Weighting of Training Samples

We studied an additional approach which handles the problem without utilizing methods of data augmentation. As shown in [15] a weighting of the loss can help to deal with imbalanced classes. In this work, the idea is to weight training samples differently. Samples with low weights are considered less important during training while those associated with high weights have stronger influence in the training process. All samples that belong to the true prediction case are considered the higher-rate samples. To the remaining majority of the training samples a lower weight is assigned. During the training procedure, this approach penalizes the network more for misclassification of the true prediction case. With this weighting it is possible to increase the influence of true prediction samples for the training of the network.

In order to incorporate the introduced weighting approach into the training procedure of the RNN, the loss values of different training samples are individually weighted by multiplying each calculated loss value with a weighting coefficient. For simplification, training examples corresponding to the true prediction case are equally weighted by a pre-defined weighting coefficient which is larger than 1.0. The loss values of other training samples remain the same.

In order to inject this weighting scheme into the weighting loss function in (3) we compare for a time step t the label of the current window y_t (corresponding to the right border of the windowed signal) with the label $y_{t+\tau}$ that should be predicted. If both labels are the same, the weight w^{TP} is set to one. Otherwise, it is set to the pre-defined value. The weighting loss in (3) is then re-written as

$$E_w(\Theta, \mathbf{X}, \mathbf{Y}) = \frac{-2}{T(T+1)} \sum_{t=1}^T t w_t^{\text{TP}} \mathbf{y}_t \log(\hat{\mathbf{y}}_t(\mathbf{X}_1 : t, \Theta)) \quad (6)$$

with w_t^{TP} denoting the weight w^{TP} for time step t .

V. EXPERIMENTS

A. Datasets

Three databases from the Ninapro project [11] were used to conduct the experiments. These databases have been used often in the field of sEMG based hand movement classification [4], [16], [17].

The experiments of the first database (DB I) are conducted for 27 subjects who were abled and healthy. The subjects performed 53 different hand movements (including rest) which were repeated 10 times for each movement except rest. The signals were recorded using 10 *MyoBock 13E200-50* electrodes (Otto Bock HealthCare GmbH, www.ottobock.com).

The second and third database (DB II and DB III) differ slightly from DB I in the experimental setup. Instead of 53 just 50 hand movements were conducted. Also, the subjects were asked to repeat the movements just 6 times. For DB II and DB III 12 *Trigno Wireless* electrodes (Delsys, Inc, www.delsys.com) were used for data acquisition. DB II includes experiments conducted by 40 abled and healthy subjects, whereas DB III contains measurements of 11 trans-radial amputated subjects.

In this work, preprocessing, feature extraction, and the splitting in training and test data were done as in [11].

B. Preprocessing and Features

As in previous works, the preprocessing followed the scheme in [18]. Firstly, the raw signals were channel-wise normalised to achieve a zero mean and unit standard deviation. The statistics used for the normalisation were estimated on training data only. Subsequently, overlapping windows ($w = 100$ ms) were extracted using a hop of 10 ms. Afterwards, each window was represented by a set of features which were calculated channel-wise. Note that the selected features are the same as the ones in previous works as well as the channel-wise normalisation of the resulting feature vectors [9], [10].

C. Error Measure for True Prediction

We have introduced strategies for dealing with the early prediction of hand movements. To evaluate their influence, it is necessary to quantify their performance in the true prediction case. Therefore, a slightly modified version of the accuracy is used. The so called true prediction accuracy is defined as $\text{acc}_{\text{TP}} = \frac{\hat{N}_{\text{TP}}}{N_{\text{TP}}}$, where N_{TP} is the number of samples corresponding to the true prediction case and \hat{N}_{TP} denotes the number of true prediction case samples that are correctly recognized by the predictor. A combined evaluation of accuracy and true prediction accuracy allows us to investigate the trade-off problem between the overall performance and the reliability in the true prediction case.

D. Experimental Results

In all experiments a single GRU cell with a state of size 256 was used. The accuracies and true prediction accuracies reported for each database individually are the average over all of the subjects of the experimental database.

Classification:

To show that the proposed approach is sufficient to handle the classification of hand movements the results achieved using 100 ms long windows are compared with the baseline systems applied on windows of length 200 ms. Since we compare with the commonly used classification the prediction offset is $\tau = -50$ ms. For DB I, DB II, and DB III we achieved accuracies of 79.3 %, 78.0 %, and 55.3 % respectively. The proposed method outperforms the best systems in [11] by 4.0 %, 2.7 %, and 9.0 % absolute as well as the state of the art methods as [19] even though we used windows of half the length. The comparison of the obtained results with the reported results for classification in [10] reveals that the

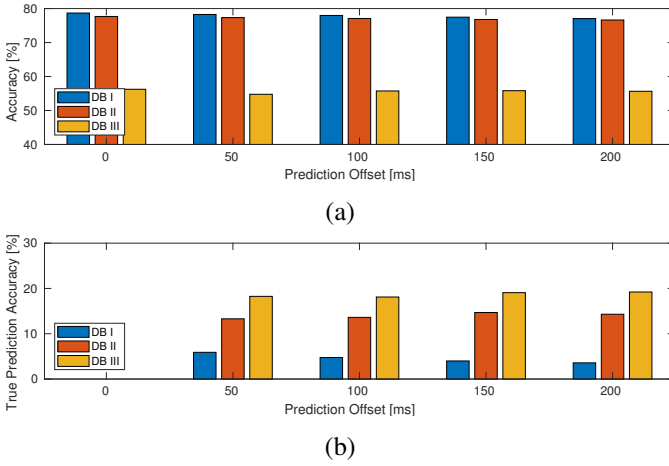


Fig. 2. Results on early prediction obtained with the RNN based network. (a) Accuracy, (b) true prediction accuracy.

results for the proposed approach are better, especially for DB III where the improvement is 1.5 %. Since the approaches only differ in their training, the proposed strategies in Section III could be helpful for improving the performance.

Early Prediction – Naive:

Fig. 2 show the results for the early prediction using none of the techniques introduced in Section IV. A prediction offset of $\tau = 0$ ms corresponds to classifying the label that belongs to the end of the window. As expected, the accuracy decreases with increasing prediction offset. But the drop of accuracy is marginal. Regarding the true prediction accuracy there is no such trend. For DB I the true prediction accuracy drops more for higher offsets. For the other databases the true prediction accuracy has a tendency to increase with increasing prediction offset. That might be due to the larger number of true prediction samples in the training and test data. Overall, the true prediction accuracy is for all cases significantly better than guessing.

Early Prediction – Oversampling:

In the following, we present the results achieved using the proposed methods for improving the performance in the true early prediction cases described in Section IV beginning with the oversampling approach (see Section IV-A). The results in Fig. 3 were obtained using an oversampling factor of 20. Since the sampling frequency of the data in DB I is too low for this experiment, this experiment was conducted using DBs II and III only. A comparison of the overall accuracies of the naive approach and the oversampling approach reveals that when using the oversampling method the accuracy is not as good as for the naive approach. The difference between them becomes worse with increasing prediction offset and can exceed 5%. Nevertheless the true prediction accuracy can be improved with the oversampling approach. In general, the effect of the oversampling becomes more obvious with increasing prediction offset. In some experiments, the true prediction accuracy can be improved by more than 10% absolute.

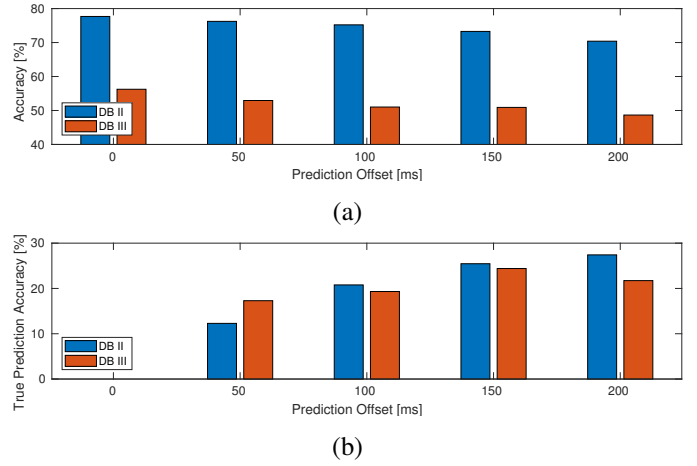


Fig. 3. Results on early prediction obtained with the RNN based network and the oversampling approach. (a) Accuracy, (b) true prediction accuracy.

Early Prediction – Weighting:

In order to evaluate the second strategy for improving the early prediction performance which uses a different weighting of the training examples as explained in Section IV-B, two experiments with weightings w^{TP} of 10 and 25, respectively, were conducted. Fig. 4 show the obtained results. Considering the weighting $w^{\text{TP}} = 10$ (see Fig. 4(a-b)) for all three databases, the overall accuracy drops often less than 1.0% compared to the results without additional methods for improving true prediction performance, except for higher predictions offsets like 150ms and 200ms where the accuracy can drop around 2.0%. However, considering the true prediction accuracy shown in Fig. 4(b), it becomes obvious, that this modification of the training process improves the performance in true prediction cases quite significantly for all three databases. The true prediction accuracy can be improved by up to 10.8% absolute. Consequently, the proposed technique is beneficial for both abled subjects as well as amputees. Considering DBs II and III, for nearly all positive prediction offsets the true prediction accuracy exceeds 20%. For DB I, the true prediction accuracy can be more than twice as good compared to the results for the naive approach. A comparison of the true prediction accuracies in Fig. 4(b) reveals that for higher prediction offsets no drop in true prediction accuracy is seen, often even slightly better true prediction accuracies are achieved. This may be counter-intuitive, but one explanation might be that the number of true prediction samples in the training set is increased by longer prediction offsets.

To evaluate the influence of the weighting w^{TP} and to validate the suitability of the approach, the experiments were repeated with $w^{\text{TP}} = 25$. The results are shown in Fig. 4(c-d). In general, the observations made are similar to the ones for $w^{\text{TP}} = 10$ but more extreme. The overall accuracies are lower than for the naive approach but the drop is often slightly bigger as for the lower weighting w^{TP} . However, in general the true prediction accuracies are remarkably higher. It is obvious that the different weighting of the training examples allows

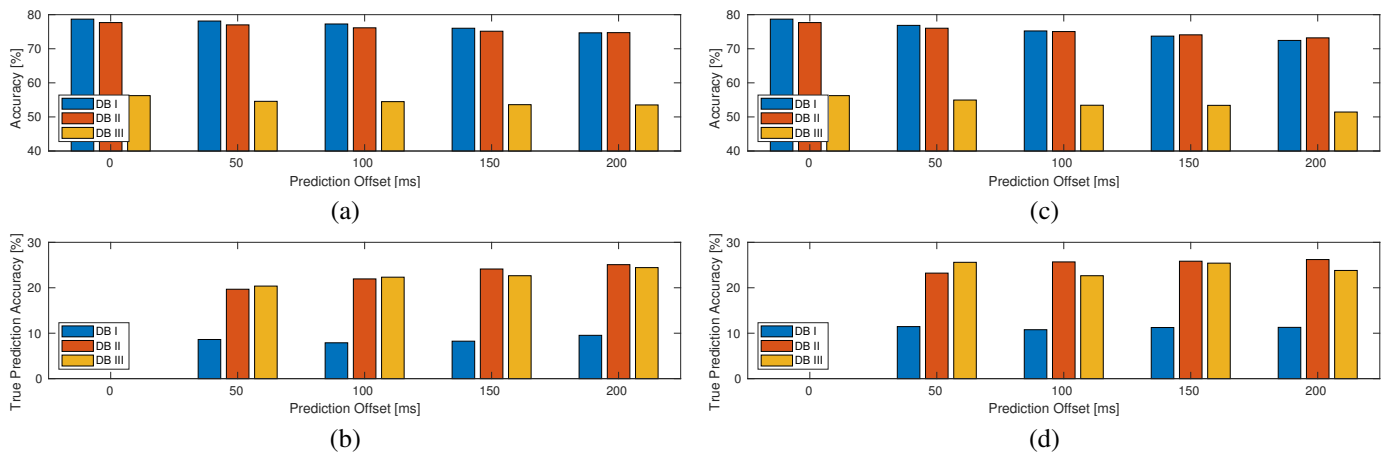


Fig. 4. Results on early prediction obtained with the RNN based network and a weighting of true prediction examples. For a weighting of $w^{TP} = 10$: (a) accuracy, (b) true prediction accuracy. For a weighting of $w^{TP} = 25$: (c) accuracy, (d) true prediction accuracy.

us to significantly improve the true prediction accuracy while causing only marginal impairment of the overall accuracy.

VI. CONCLUSION

We showed in this work that the used RNN based classification approach in combination with the proposed modification of the training procedure outperform state of the art systems for hand movement classification by far. The approach works for both abled subjects and amputees. Also, promising results for the early prediction task could be achieved. The performance on early prediction was significantly improved by relatively simple methods. Both, the oversampling strategy and the weighting of the training examples cause directly or indirectly a change of the training data distribution in a way that the true prediction case becomes more important. Since both strategies lead to good results, it is likely that a higher number of hand movement changes in the dataset would cause an enhancement of the performance in true prediction case. A combination of differently trained early predictors could be a promising approach as well.

REFERENCES

- [1] C. Castellini and P. van der Smagt, "Surface EMG in advanced hand prosthetics," *Biol. Cybern.*, vol. 100, no. 1, pp. 35–47, 2009.
- [2] A. H. Al-Timemy, G. Bugmann, J. Escudero, and N. Outram, "Classification of finger movements for the dexterous hand prosthesis control with surface electromyography," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 3, pp. 608–618, 2013.
- [3] H. Daley, K. Englehart, L. Hargrove, and U. Kuruganti, "High density electromyography data of normally limbed and transradial amputee subjects for multifunction prosthetic control," *J. Electromyogr. Kinesiol.*, vol. 22, no. 3, pp. 478 – 484, 2012.
- [4] M. Atzori, C. M., and H. Müller, "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands," *Front. Neurobot.*, vol. 10, no. 9, 2016.
- [5] A. Gijsberts, M. Atzori, C. Castellini, H. Müller, and B. Caputo, "The movement error rate for evaluation of machine learning methods for sEMG-based hand movement classification," *IEEE Trans. Neural Systems Rehab. Eng.*, vol. 22, no. 4, pp. 735 – 744, 2014.
- [6] M. P. Mobarak, R. M. Guerrero, J. M. G. Salgado, and V. L. Dorr, "Hand movement classification using transient state analysis of surface multi-channel EMG signal," in *2014 Pan American Health Care Exchanges*, 2014, pp. 1–6.
- [7] Y. Zhang, G. Wang, C. Teng, Z. Sun, and J. Wang, "The analysis of hand movement distinction based on relative frequency band energy method," *BioMed Research International*, vol. 2014, Nov. 2014.
- [8] L. H. Smith, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, "Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 2, pp. 186–192, 2011.
- [9] P. Koch, H. Phan, M. Maass, F. Katzberg, and A. Mertins, "Early prediction of future hand movements using sEMG data," in *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2017, pp. 54–57.
- [10] —, "Recurrent neural network based early prediction of future hand movements," in *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2018, *accepted*.
- [11] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. Mittaz Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Sci. Data*, vol. 1, no. 140053, 2014.
- [12] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.
- [13] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [14] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–13.
- [15] H. Phan, M. Krawczyk-Becker, T. Gerkmann, and A. Mertins, "Weighted and multi-task loss for rare audio event detection," in *Proc. IEEE Int. Conf. Acoust. Speech. Signal. Process.*, 2018, pp. 336–340.
- [16] A. Krasoulis, S. Vijayakumar, and K. Nazarpour, "Evaluation of regression methods for the continuous decoding of finger movement from surface EMG and accelerometry," in *Proc. International IEEE/EMBS Conference on Neural Engineering*, 2015, pp. 631–634.
- [17] M. AbdelMaseeh, T.-W. Chen, and D. W. Stashuk, "Extraction and classification of multichannel electromyographic activation trajectories for hand movement recognition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 6, pp. 662–673, 2016.
- [18] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Engineering*, vol. 50, no. 7, pp. 848–854, 2003.
- [19] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface EMG images," *Sci. Rep.*, vol. 6, no. 36571, 2016.