Recurrent Network based Automatic Detection of Chronic Pain Protective Behavior using MoCap and sEMG data

Chongyang Wang[1], Temitayo A. Olugbade[1], Akhil Mathur[1], Amanda C. De C. Williams[1], Nicholas D. Lane[2], Nadia Bianchi-Berthouze[1]
[1]UCL interaction centre, University College London, {chongyang.wang.17, temitayo.olugbade.13, akhil.mathur.17, amanda.williams, nadia.berthouze}@ucl.ac.uk
[2]Department of Computer Science, University of Oxford, nicholas.lane@cs.ox.ac.uk

ABSTRACT

In chronic pain physical rehabilitation, physiotherapists adapt exercise sessions according to the movement behavior of patients. As rehabilitation moves beyond clinical sessions, technology is needed to similarly assess movement behaviors and provide such personalized support. In this paper, as a first step, we investigate automatic detection of protective behavior (movement behavior due to pain-related fear or pain) based on wearable motion capture and electromyography sensor data. We investigate two recurrent networks (RNN) referred to as stacked-LSTM and dual-stream LSTM, which we compare with related deep learning (DL) architectures. We further explore data augmentation techniques and additionally analyze the impact of segmentation window lengths on detection performance. The leading performance of 0.815 mean F1 score achieved by stacked-LSTM provides important grounding for the development of wearable technology to support chronic pain physical rehabilitation during daily activities.

CCS CONCEPTS

• Applied computing → Life and medical sciences → Health informatics • Human-centred computing → Ubiquitous and mobile computing

KEYWORDS

Physical rehabilitation, Affective behavior, Recurrent networks

ACM Reference format:
Chongyang Wang, Temitayo A. Olugbade, Akhil Mathur, Amanda C. De C. Williams, Nicholas D. Lane, Nadia Bianchi-Berthouze. 2019. Recurrent Network based Automatic Detection of Chronic Pain Protective Behavior using MoCap and sEMG data. In *Proceedings of International Symposium on Wearable Computers (ISWC'19). ACM, London, UK, 4 pages.* https://doi.org/10.1145/1234567890

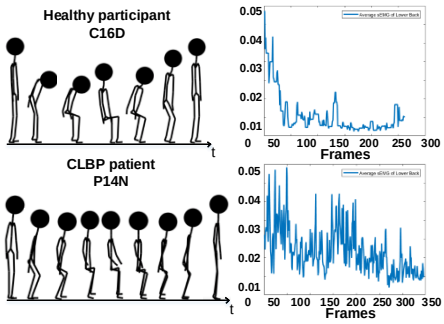

Figure 1: MoCap and sEMG sequences of CLBP patient (showing protective behavior) and healthy participant performing stand-to-sit-to-stand. The average upper envelope of sEMG data from the lower back is presented.

1 Introduction

Physical rehabilitation is an important part of the management of chronic pain (CP), where pain associated with dysfunction in the nervous system (rather than tissue damage) leads to impaired engagement in everyday physical activities [1, 2]. Maladaptive strategies or protective behaviors (e.g. rigidity or stiffness in movement, use of support) [3] emerge from fear and low self-efficacy of/for the movements that are essential to these activities [4, 5, 33, 34, 36]. To address these underlying factors, physiotherapists adapt the type and amount of feedback and the forms of activities prescribed based on their observations of the strategies a patient uses during physical rehabilitation sessions in pain management programs [6, 35]. As the rehabilitation is moving from clinical settings to home-based self-management, technology should be able to provide similar services by detecting these behaviors [35]. Wearable body sensing technology provides unique opportunities for physical rehabilitation in that: i) it can enable personalized, real-time feedbacks to be accessible to patients outside of clinical settings; and ii) it can be used ubiquitously as physical rehabilitation at home comprise unconstraint daily activities. To fully realize the potential of body sensing technology for such everyday settings, it is important to understand the feasibility of automatic assessment of movement behaviors in people with chronic pain. In this paper, we address this with an investigation of automatic detection of protective behaviors based on data captured using

wearable motion capture (MoCap) and surface electromyography (sEMG) sensors (see Figure 1). Our contributions can be summarized as:

1. We investigate the possibility of using deep learning to detect protective behavior within activity segments from a variety of activities.
2. We explore two possible augmentation methods for MoCap and sEMG data to enable deep learning, namely, jittering and random discarding.
3. We analyze the impact of the sliding window segmentation with different window lengths on the modeling across activity types.

2 Related Works

The majority of the work done on automatic detection of pain behavior (including protective behavior) has been on automatic differentiation of people with CP from healthy control participants, as in the studies of [13, 14, 15] on lower back and neck CP. Protective behavior is also seen as a cue of low self-efficacy, and [11] proposed to use feature-engineering methods to characterize it. [9, 10, 35] further provides evidence that low-cost body sensing technology can enable the detection of pain related experiences in functional activities. One study along this line is by Aung et al. [16], where a new dataset called EmoPain dataset [8] is used, which remains the only chronic pain body movement dataset available. From each activity instance, they computed the range of each joint angle, the mean energy, and the mean of the upper-envelope of rectified sEMG data for the 4 bilateral upper and lower back muscle activities. Random forest was used on these features to detect protective behavior during different activities, which achieved mean square errors between 0.019 to 0.034 (mean=0.04, std=0.16). A limitation of these works is the use of activity-dependent feature and non-temporal approaches to model the behaviors, which lacks generalization ability to other activities.

Findings in human activity recognition (HAR) literature further point to the efficacy of DL networks for this direction. To name a few, [17] used a bi-directional LSTM (Bi-LSTM) to classify physical activities. They obtained mean F1 scores of 0.75 and 0.94 in the Opportunity [18] and PAMAP2 [19] datasets respectively based on hold-out validation. In another work, [20] achieved mean F1 scores of 0.73 and 0.85, based on hold-out validation, respectively on the same datasets using an ensemble of two-layer LSTM networks with dropouts after each layer. This method further led to mean F1 score of 0.92 on the Skoda dataset [21]. Another study [22] used a stack of three convolutional layers (with max pooling after the first two), an LSTM layer, and a dense layer (with softmax activation) to classify physical activities in the Opportunity dataset and the Skoda dataset. [17] further used a three-layer LSTM network to automatically detect freezing behavior in people with Parkinson's while they performed walking, they obtained mean F1 score of 0.76 based on data from the Daphnet Gait [23] dataset. Here the freezing behavior would usually interrupt the activity rather than modifying the way it is performed. Hence, the recognition of freezing behavior is deemed as the recognition of a new activity type (freezing).

In this paper, we build on the HAR literatures but to capture protective behavior within activities rather than the activity itself. We also aim to understand if such behavior can be detected independently of the type of activity performed within a pool of five daily activities considered demanding by people with CP. We choose LSTM-based architectures as they can better capture the dynamic aspect for automatic detection of protective behavior and compare them with convolution-based networks.

3 Building Recurrent Networks with LSTM

Given the inherent dynamic and temporal nature of MoCap and sEMG data, we consider RNNs for our task. At the core of any RNN architecture is the processing unit. One of the most widely applied unit in RNNs is the LSTM [27] which solved the gradient vanishing problem which traditional RNNs face in backpropagation over a long period. Every LSTM unit updates its internal states based on previous information. To extract long-term temporal information in the direction natural to the expression of protective behavior in physical activities, we choose forward information passing for our architecture. Nevertheless, we experiment with the bi-directional information processing architecture (Bi-LSTM) to explore the difference. The LSTM unit that we use is the vanilla variant without peephole connection [28]. In this paper, the first RNN we applied is built by stacking such LSTM layers as shown in Figure 2 (a). An experiment on the number of LSTM units and layers was conducted and we found that the optimal combination is a 3-layer network with 32 hidden units in each. In this paper, we refer to this network as stacked-LSTM.

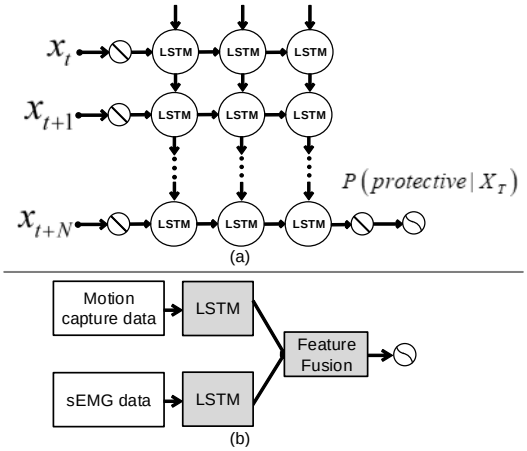

Figure 2: (a) The applied RNN with LSTM layers stacked; (b) The dual-stream LSTM.

The input of a LSTM unit is the current input data $\boldsymbol{x_t}$, previous hidden state $\boldsymbol{h_{t-1}}$ and the previous cell state $\boldsymbol{c_{t-1}}$, while the output is the current hidden state $\boldsymbol{h_t}$ and cell state $\boldsymbol{c_t}$. Here, the input data $\boldsymbol{x_t}$ is equal to a single sample in the frame at timestep t. Within each LSTM unit, the Input Gate with output $\boldsymbol{i_t}$, Forget Gate with output $\boldsymbol{f_t}$, Output Gate with output $\boldsymbol{o_t}$ and Cell Gate with output $\tilde{\boldsymbol{c}}_{\boldsymbol{t}}$ are updated as:

$$\boldsymbol{\varphi_t} = \binom{\sigma}{tanh}(\boldsymbol{W_{x*}x_t} + \boldsymbol{W_{h*}h_{t-1}} + \boldsymbol{b_*}) \quad (1)$$

$$\boldsymbol{c_t} = \boldsymbol{f_t} \odot \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \odot \tilde{\boldsymbol{c}}_{\boldsymbol{t}} \ , \ \boldsymbol{h_t} = \boldsymbol{o_t} \odot \tanh(\boldsymbol{c_t}) \quad (2)$$

$$\sigma(x) = (1 + e^{-x})^{-1} \ , \ \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (3)$$

where $\boldsymbol{\varphi_t} \in \{\boldsymbol{i_t}, \boldsymbol{f_t}, \boldsymbol{o_t}, \tilde{\boldsymbol{c}}_{\boldsymbol{t}}\}$ with the $tanh$ function only used for $\tilde{\boldsymbol{c}}_{\boldsymbol{t}}$, $\odot$ denotes the element-wise multiplication, $\boldsymbol{W_{x*}}$ **and** $\boldsymbol{W_{h*}}$ are weight matrices and $\boldsymbol{b}_*$ are bias vectors.

The processing at next timestep $t + 1$ takes the current output $\boldsymbol{c_t}$ and $\boldsymbol{h_t}$ to iterate the same computation. For the current input frame $\boldsymbol{X_T} = [\boldsymbol{x_t}, \boldsymbol{x_{t+1}}, \dots, \boldsymbol{x_{t+N}}]$ (generated by sliding-window segmentation with length of $N + 1$), given the last output hidden state $\boldsymbol{h_{t+N}}$ from the last LSTM layer, the class probability $\boldsymbol{p} = [p_1, \dots, p_K]$ where K denotes the number of classes (in our case $K = 2$) and the final label prediction Y are computed as:

$$\boldsymbol{p} = softmax(\boldsymbol{W_{hK}} \boldsymbol{h_{t+N}} + \boldsymbol{b_K}) \ , \ Y = \underset{[1,K]}{arg\max}(\boldsymbol{p}) \quad (4)$$

where $\boldsymbol{W_{hK}}$ and $\boldsymbol{b_K}$ are the learnable weight matrix and bias vector of the last fully-connected layer in the network.

Based on stacked-LSTM network, a variant can be created by using two streams of such network, where the MoCap and sEMG data are processed separately and combined at feature level. We refer to this architecture as dual-stream LSTM which is shown in Figure 2 (b). Each LSTM block represents the temporal processing part similar to stacked-LSTM.

4 Data Preparation

The EmoPain dataset [8] contains MoCap and sEMG data collected from 26 healthy participants and 22 chronic lower-back pain (CLBP) patients doing typically feared activities. We use this dataset to learn patterns of protective behavior exhibited by CLBP patients during different activities. Four expert raters (2 physiotherapists and 2 clinical psychologists) labelled the data by indicating the starting and ending points of protective behavior and its type. The data from healthy participants were all labelled as non-protective. Whilst the dataset contains 22 patients, 4 patients were left out because of errors in their sEMG data recordings. To avoid biasing the model towards healthy participants, only 12 healthy people were randomly used.

Figure 1 shows avatar examples of protective and non-protective behaviors from the EmoPain dataset. Differently from the top healthy participant, the stick figures of the CLBP patient do not bend the trunk but exploit the leg muscles to lower him/herself to the seat, a strategy further facilitated by twisting the trunk to minimize the use of the left (possibly painful) part of the back.

The five activities we chose to model from the EmoPain dataset are bending, stand-on-one-leg, sit-to-stand, stand-to-sit and reach-forward, while the rest represents transition movements like standing still, sitting still and walking around. In total, we have 46 activity instances from the CLBP and healthy subjects, where each instance lasts 10 minutes or so. Additionally, for the MoCap data, we compute angles and energies, instead of using the raw Euclidean positions. This is because the angle is invariant to the change of joint position in the movement space and also to provide better representation with smaller dimensionality of body movement [32]. At each timestep, a total of 13 angles are calculated in 3D space as suggested in [8] based on the 26 anatomical points to describe the local movements of the body, where the energies are the square of their respective angular velocities. For the muscle activity, we use the upper envelope of the rectified sEMG to provide smooth and denoised representation of the raw data.

4.1 Data Segmentation

For both the training and testing set, a sliding-window segmentation method [30] is applied to generate frames of continuous portions of the data instances. The size of the sliding-window was selected upon an in-depth analysis based on the different activity types (discussed in section 5.3). Using the best parameter which emerged from this analysis (window length of 3 seconds with overlapping ratio of 75%), the segmentation done on the 46 movement instances led to 2646 frames.

4.1.1 Ground truth computation. Due to the limited data size, for the five categories of protective behavior (e.g., guarding, hesitation, support, abrupt motion and stimulation) [8], we combined them into one special class named protective behavior. Such practice lead to a binary detection task discriminating between protective and non-protective behavior. To compensate for the disagreement between the expert raters, we define the ground truth of each frame using the majority-voting rule: a frame is labelled as protective if no less than 50% percent of the included samples were marked as protective by at least 2 raters.

4.2 Data Augmentation

To enable the training of DL networks on a comparatively small dataset we apply two different data augmentation methods aimed to simulate real-life situations as patients wear the sensors: i) *Jittering [12]*, which is to simulate the signal noise that may exist during data capturing in real life. We create normal Gaussian noise under three standard deviations of 0.05, 0.1, 0.15 and globally add each to the original data; ii) *Random Discarding (similar to cropping [12])*, which is to simulate unexpected data loss at some points during the recording. We randomly set the data at some timesteps as well as body parts to 0 with selection probabilities of 5%, 10% and 15%.

All these data augmentation methods do not change the temporal order of the data neither the movements to a noticeable degree. Therefore, the ground truth labels stay unchanged. After testing the two data augmentation methods with stacked-LSTM on the entire dataset, we found that the performance was improved (about 18% higher mean F1) with a combination of them. Consequently, we use the combination of these two approaches as the default augmentation method for all the DL architectures used for comparisons. The number of frames created after using such combined augmentation method is 18,653 (before is 2646), where 11373 frames are labelled as non-protective (from both healthy participants and patients) with 7280 frames as protective (only from the patients).

5 Experiments

For comparison purposes, all the neural networks used in our experiments employed the Adam [29] optimizer to update the

weights. During training, we used a mini-batch size of 20 and a fixed learning rate of 0.001. The deep learning methods are implemented using Keras with TensorFlow backend. The hardware used is a workstation with Intel i7 8700K CPU. No GPU acceleration is employed.

Aside from the two architectures described earlier, namely stacked-LSTM and dual-stream LSTM, we use a CNN, a Conv-LSTM and a Bi-LSTM proposed in HAR scenarios [24, 22, 17] for comparison. An optimization experiment was conducted to find the optimal hyper-parameter settings of these methods: i) For stacked-LSTM, three LSTM layers each with 32 hidden units followed by a dropout layer with probability of 0.5 are used; ii) For dual-stream LSTM, two sets of 3 LSTM layers each with 24 hidden units and 8 hidden units followed by dropout layer with probability of 0.5 are used respectively for the MoCap and sEMG streams; iii) For CNN [24], three convolutional layers each with 10 kernels of size 1×10 and followed by 1×2 max-pooling layer are used, and a softmax layer in the end is used for classification; iv) For Conv-LSTM [22], the architecture is used with 10 kernels of size 1×10 in each convolutional layer followed by max-pooling of size 1×2, and the number of hidden units of each LSTM layer is set to 32; v) For Bi-LSTM [17], 3 bi-directional LSTM layers with 16 hidden units in each are used. Here we need to mention that, the reason we found kernel size of 1×10 is better than the popular one (e.g., 1×5) could be that, given the input size of $N \times 180 \times 30$ (N is the number of samples), such temporal length (180) demand larger kernel size.

Given that in our scenario the detection of both protective and non-protective behaviors is similarly important, we report mean F1 score as the metric for both outcomes

5.1 Comparison of DL Architectures

The comparison results with leave-one-subject-out validation (LOSO) are shown in Table 1. We can see that stacked-LSTM achieves the best F1 score of 0.815 and is about 12% and 5% respectively higher than CNN and Conv-LSTM. Here we also conducted a repeated measures ANOVA between these methods and an effect of methods on performances is found: $F(0.651, 4.054)=6.311$, $p<0.001$, $\mu^2=0.179$. Post-hoc paired t-tests with Bonferroni corrections further show that stacked-LSTM performs significantly better than CNN and Conv-LSTM ($p=0.003$ and 0.032). However, significant differences are not found between Bi- LSTM, dual-stream LSTM and stacked-LSTM, which suggest that the variants created within the LSTM framework lead to better results than convolution-based ones but the different types of LSTM architecture used does not contribute to the effect.

5.2 DL model and Human Experts Agreement

For the 18 folds in the LOSO validation where testing subjects are patients, we further compute two-way mixed, absolute agreement intraclass correlations (ICCs) to compare the level of agreement between the ground truth labels we generated by majority-voting and the stacked-LSTM with what between the expert raters. The ICC is a standard method for computing interrater agreement [25] and the two-way mixed model was used to account for the rater effect [31]. We found ICC = 0.215

	Metrics		
DL architectures	Acc	Mean F1	p-value (<0.05)
CNN	0.7736	0.697	0.003
Conv-LSTM	0.7913	0.767	0.032
Bi-LSTM	0.8033	0.794	>0.05
Dual-LSTM	0.8035	0.795	>0.05
Stacked-LSTM	**0.8686**	**0.815**	-

Table 1: Comparison results for the DL architectures.

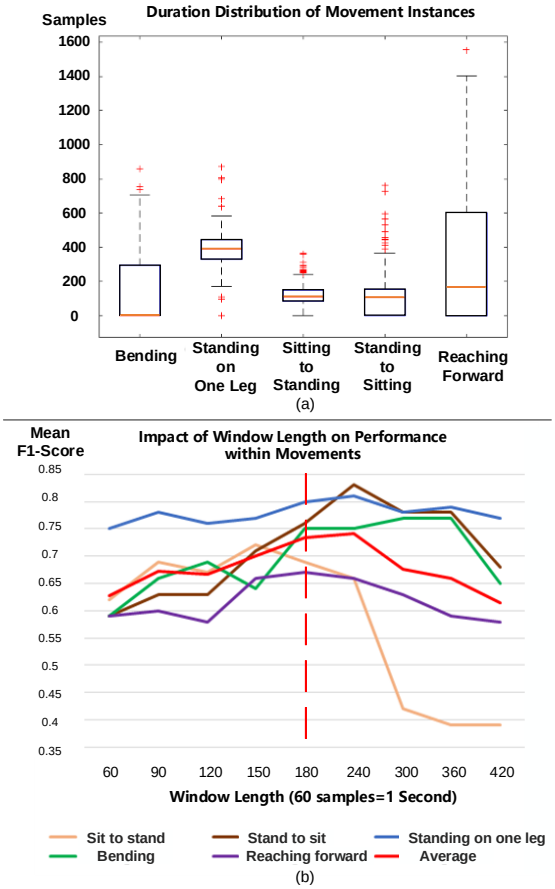

Figure 3: (a) The duration distribution of activity instances in EmoPain dataset. 60 samples are equal to 1 second; (b) The detection performances across different window lengths within different activities, the dotted-line is at window length of 3s.

(single measures) and 0.523 (average measures) with p=4.3e-130, between the raters, and ICC=0.568 (single measures) and 0.724 (average measures) with p=3.1e-159, between stacked-LSTM and the ground truth. This finding suggests that stacked-LSTM reaches excellent level of agreement with the average expert rater (the ground truth created by majority-voting), which aligns with the goal of our modelling.

5.3 Segmentation Analysis

The sliding-window segmentation approach used for the comparison of DL architectures is based on a window length of 3

seconds with overlapping ratio of 75%. The boxplot in Figure 3 (a) shows the duration of activity instances in the EmoPain dataset. As is shown, there are strong differences between the durations of the different activity types and even between instances within the same activity type. This is possibly due to people's different physical and psychological (fear, anxiety) capabilities. [26] suggested that the window length need to be adjusted to different types of activity. Consequently, we use the stacked-LSTM to run experiments on different activity types separately with frames generated by different window lengths while using a fixed overlapping ratio of 75%. Results (Figure 3 (b)) show that the window length affects the detection result (mean F1-score) to some degree for most activity types. Except for the different dynamic characteristics of activities, one obvious explanation is that the size of training data generated is smaller with a longer sliding-window. Still, the best window of 3s was tested to reach good performance across activities, as shown in this figure and in section 5.1 when the recognition of protective behavior is done independently of the activity performed.

6 Conclusions

In this paper, we investigate the possibility of automatically detecting frames of protective behavior in people suffering from CLBP during emotionally and physically demanding activities. The aim is to develop movement- and muscle activity-recognition wearable technology that can be used to personalize feedback and support planning of self-directed (i.e. without a clinician) and long-term ubiquitous physical rehabilitation. Using the EmoPain dataset (MoCap and sEMG data), we show that the recurrent network with LSTM units referred to as stacked-LSTM performs much better than the CNN-based models: mean F1score of 0.815, while leading to excellent agreement with the average expert raters. Another proposed model called dual-stream LSTM performed similarly to stacked-LSTM, which showed the good potential of using better architecture that pay respect to different data types. Rather than using the raw data, we make use of very low-level features. These features together with the data augmentation methods make it more feasible to model the protective behavior with basic deep learning architectures. We additionally provide evidence of the impact of sliding-window lengths on detection performance and we suggest that the choice be based on some knowledge of the dataset. The results suggest that this parameter is affected by the duration of the movement but also by the complexity of the movement as the temporal characteristic varies. Still, generalization can be achieved without a critical loss in performance.

7 Acknowledgement

The project was supported by the Future and Emerging Technologies (FET) Proactive Programme H2020-EU.1.2.2 (Grant agreement 824160; EnTimeMent) and Chongyang Wang is supported by the UCL Overseas Research Scholarship (ORS) and Graduate Research Scholarship (GRS).

REFERENCES

[1] Breivik, H., et al. (2006). Survey of chronic pain in Europe: Prevalence, impact on daily life, and treatment. European Journal of Pain, 10 (4), 287.
[2] Tracey, I. et al. (2009). How Neuroimaging Studies Have Challenged Us to Rethink: Is Chronic Pain a Disease? Journal of Pain, 10 (11), 1113-1120.
[3] Keefe, F. J. et al. (1982). Development of an observation method for assessing pain behavior in chronic low back pain patients. Behavior Therapy.
[4] Vlaeyen, J. W. S. et al. (2000). Fear-avoidance and its consequences in chronic musculoskeletal pain: A state of the art. Pain, 85 (3), 317-332.
[5] Vlaeyen, J. W. S., et al. (2016). The experimental analysis of the interruptive, interfering, and identity-distorting effects of chronic pain. Behaviour Research and Therapy, 86, 23-34.
[6] Singh, A., et al. (2014). Motivating People with Chronic Pain to do Physical Activity: Opportunities for Technology Design. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), 2803-2812.
[7] Singh, A., et al. (2016). Go-with-the-Flow: Tracking, Analysis and Sonification of Movement and Breathing to Build Confidence in Activity Despite Chronic Pain. Human–Computer Interaction, 31 (3-4), 335-383.
[8] Aung, MSH, et al. (2016). The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal EmoPain dataset. IEEE Transactions on Affective Computing, 7 (4), 435-451.
[9] Olugbade, T. A., et al. (2014). Bi-Modal Detection of Painful Reaching for Chronic Pain Rehabilitation Systems. Proceedings of the 16th International Conference on Multimodal Interaction (ICMI), 455–458.
[10] Olugbade, T. A., et al. (2015). Pain Level Recognition using Kinematics and Muscle Activity for Physical Rehabilitation in Chronic Pain. International Conference on Affective Computing and Intelligent Interaction (ACII), 243-249.
[11] Olugbade, T. A., et al. (2018). Human Observer and Automatic Assessment of Movement Related Self-Efficacy in Chronic Pain: from Movement to Functional Activity. IEEE Transactions on Affective Computing.
[12] Um, Terry Taewoong et al. (2017). Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. arXiv preprint arXiv:1706.00527.
[13] Watson, P. J., et al. (1997). Evidence for the Role of Psychological Factors in Abnormal Paraspinal Activity in Patients with Chronic Low Back Pain. Journal of Musculoskeletal Pain, 5 (4), 41-56.
[14] Ahern, D. K., et al. (1988). Comparison of lumbar paravertebral EMG patterns in chronic low back pain patients and non-patient controls. Pain, 34(2), 153-160.
[15] Grip, H., et al. (2003). Classification of Neck Movement Patterns Related to Whiplash-Associated Disorders Using Neural Networks. IEEE Transactions on Information Technology in Biomedicine, 7 (4), 412-418.
[16] Aung, MSH, et al. (2014). Automatic recognition of fear-avoidance behavior in chronic pain physical rehabilitation. Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare (ICPCTH), 158-161.
[17] Hammerla, Nils Y., et al. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. Proceedings of the 25th International Joint Conference on Artificial Intelligence.
[18] Chavarriaga, Ricardo, et al. (2013). The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters, 34 (15), 2033-2042.
[19] Reiss, Attila et al. (2012). Introducing a new benchmarked dataset for activity monitoring. 16th International Symposium on Wearable Computers (ISWC), 108-109.
[20] Guan, Yu et al. (2017). Ensembles of deep lstm learners for activity recognition using wearables. Proceedings of ACM on Interactive, Mobile, Wearable, Ubiquitous Technologies (IMWUT), 1 (2), 11.
[21] Daniel Roggen, et al. (2008). Wearable activity tracking in car manufacturing. IEEE Pervasive Computing, 1 (2), 42-50.
[22] Morales, Francisco Javier Ordóñez et al. (2016). Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. 20th International Symposium on Wearable Computers (ISWC), 92-99.
[23] Bachlin, Marc, et al. (2009). Potentials of enhanced context awareness in wearable assistants for Parkinson's disease patients with the freezing of gait syndrome. International Symposium on Wearable Computers (ISWC), 123-130.
[24] Rad, Nastaran Mohammadian et al. (2016). Applying deep learning to stereotypical motor movement detection in autism spectrum disorders. 16th International Conference on Data Mining Workshops (ICDMW), 1235-1242.
[25] McGraw, K. O., et al. (1996). Forming inferences about some intraclass correlation coefficients. Psychological methods, 1 (1), 30.
[26] Huynh, Tâm, et al. (2007). Scalable recognition of daily activities with wearable sensors. International Symposium on Location-and Context-Awareness (LoCA), 50-67.

[27] Hochreiter, Sepp et al. (1997). Long short-term memory. Neural computation, 9 (8), 1735-1780.
[28] Greff, Klaus, et al. (2017). LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28 (10), 2222-2232.
[29] Kingma, Diederik P et al. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[30] Andreas Bulling et al. (2014). A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys, 46 (3), 33.
[31] Hallgren KA. (2012). Computing inter-rater reliability for observational data: an overview and tutorial. Tutorials in quantitative methods for psychology, 8 (1), 23.
[32] Falco, Pietro et al. (2017). A human action descriptor based on motion coordination. IEEE Robotics and Automation Letters, 2 (2), 811-818.
[33] Asghari, A et al. (2017). Pain self-efficacy beliefs and pain behaviour: A prospective study. Pain, 94 (1), 85-100.
[34] Woby, S. R. et al. (2007). Self-efficacy mediates the relation between pain-related fear and outcome in chronic low back pain patients. European Journal of Pain, 11 (7), 711-718.
[35] Olugbade, T. A et al. (2019). How Can Affect Be Detected and Represented in Technological Support for Physical Rehabilitation? ACM Transactions on Computer-Human Interaction, 26 (1), 1.
[36] Olugbade Temi et al. The relationship between guarding, pain, and emotion. PAIN Report, 2019. (to appear)