Automatic Detection of Protective Behavior in Chronic Pain
Physical Rehabilitation: A Recurrent Neural Network Approach

CHONGYANG WANG, University College London, London, United Kingdom
TEMITAYO A. OLUGBADE, University College London, London, United Kingdom
AKHIL MATHUR, University College London, London, United Kingdom
AMANDA C. DE C. WILLIAMS, University College London, London, United Kingdom
NICHOLAS D. LANE, Oxford University, Oxford, United Kingdom
NADIA BIANCHI-BERTHOUZE, University College London, London, United Kingdom

In chronic pain physical rehabilitation, physiotherapists adapt movement to patients’ current performance especially based on the expression of protective behavior, gradually exposing them to feared but harmless and essential everyday movements. As physical rehabilitation moves outside the clinic, physical rehabilitation technology needs to automatically detect such behaviors so as to provide similar personalized support. In this paper, we investigate the use of a Long Short-Term Memory (LSTM) network, which we call Protect-LSTM, to detect events of protective behavior, based on motion capture and electromyography data of healthy people and people with chronic low back pain engaged in five everyday movements. Differently from previous work on the same dataset, we aim to continuously detect protective behavior within a movement rather than overall estimate the presence of such behavior. The Protect-LSTM reaches best average F1 score of 0.815 with leave-one-subject-out (LOSO) validation, using low level features, better than other algorithms. Performances increase for some movements when modelled separately (mean F1 scores: bending=0.77, standing on one leg=0.81, sit to stand=0.72, stand to sit=0.83, reaching forward=0.67). These results reach excellent level of agreement with the average physiotherapists’ ratings. As such, the results show clear potential for in-home technology supported affect-based personalized physical rehabilitation.

CCS Concepts: • Applied computing → Life and medical sciences → Health informatics; • Human-centred computing → Human Computer Interaction (HCI) → HCI design and evaluation methods; • Human-centred computing → Ubiquitous and mobile computing.

KEYWORDS
Physical activity, Chronic pain, Motion capture, Electromyography, LSTM, Affective behaviour

ACM Reference format:
Chongyang Wang, Temitayo A. Olugbade, Akhil Mathur, Amanda C. de C. Williams, Nicholas D. Lane, Nadia Bianchi-Berthouze. 2019. Automatic Detection of Protective Behavior in Chronic Pain Physical Rehabilitation: A Recurrent Neural Network Approach. PACM Interact. Mob. Wearable Ubiquitous Technol. Volume 0, issue 0, Article 0 (Month 2019), pages. https://doi.org/*

1 INTRODUCTION

Body sensing technology provides new possibilities for physical rehabilitation as it makes it accessible outside of clinic settings and enables personalized real-time feedback for patients. In this paper, we address the possibility of the augmentation of such technology to deal with psychological factors in long-term conditions such as chronic pain (CP). Physical rehabilitation is an important part of the management of CP, which is a condition where pain associated with dysfunctional changes in the nervous system persists and leads to reduced engagement in everyday physical activities despite lack of injury or tissue damage [1][2]. According to the fear-avoidance theory, reduced engagement and other maladaptive strategies (collectively referred to as ‘pain behaviors’), such as protective behaviors [3], are a result of fear of pain, movement, or injury due to wrong association of harmless movement with pain [4][5]. Such behaviors lead to further debilitation and lack of engagement in valued activities, e.g. employment or social life [1][4].
During physical rehabilitation sessions in pain management programs, physiotherapists adapt the type of feedback, the amount of support, and even the movements according to the protective behaviors that a patient exhibits [6] [7]. Their aim is to gradually expose patients to feared movements to reduce anxiety that could lead to increases in pain and avoidance. As physical rehabilitation for CP moves from clinic settings to self-management in the home and everyday life, physical rehabilitation technology needs to be capable of detecting these behaviors to provide such affect-based personalized support and movement plans [8].

In this paper, we investigate the possibility of using body movement and muscle activity data recorded using motion capture and surface electromyography (sEMG) sensors to detect protective behaviors to enable the provision of personalized feedback and strategy support to help people with CP, especially chronic lower-back pain (CLBP) to regain confidence when engaging in feared movement. This work aims to advance the limited literature on automatic detection of pain-related behavior [10] [11] [12] [13] by moving towards a more continuous detection of pain behavior within a movement so as to detect what part of the movement is feared by the patient and inform the personalization functions. In the future, prospective users would be able to have access to such technology to receive personalized rehabilitation treatment and everyday guidance at home.

The modeling of protective behavior in the context of chronic pain pose a number of research challenges.

- It is difficult to collect large-scale datasets from people with CP due to ethic and subjective factors. However, state-of-the-art machine learning techniques are data hungry and do not perform well with small-scale datasets. How do we increase the size of the CP training data, without conducting expensive data collection studies?
- Prior works have employed shallow learning techniques to model protective behavior in CP contexts. However, currently deep learning methods are prominently used for human activity recognition and other human sensing tasks. We ask, do deep neural networks provide any significant advantage over shallow models in this previously unexplored topic of protective behavior detection? Moreover, how do we design a data processing pipeline to make existing CP datasets compatible with various neural network architectures.
- How do we choose a deep learning architecture that fits this problem setting, including the choice of number of layers and hidden units?

In this paper:

- For the first time, our results show that it is possible to use a recurrent neural network such as a LSTM network to detect protective behavior with an accuracy of 81.5% in leave-one-subject-out (LOSO) validation. Our model outperforms all other tested baselines of shallow and deep models.
- The best model reaches excellent level of agreement with the average expert rater based on four experts (physiotherapists and psychologist)’ ratings.
- We show that it is possible to enhance the dataset from 2569 frames to 18K frames using our data augmentation techniques, which in turn increase the detection accuracy by 15.5% in LOSO validation.
- We provide a careful and systematic analysis of various hyperparameters and how they impact the detection accuracy from around chance level to more than 70%.

In summary, we makes the following contributions:

- We explore the use of recurrent neural network architectures with Long Short-Term Memory (LSTM) layers for the detection of protective behavior within frames of multiple movement instances, based on the sequence of motion capture and sEMG data within these frames. Unlike previous approaches, our models are movement type independent (i.e., generalize across movements) and we extend the modeling to a wider set of pain behavior.
- We present a number of padding and data augmentation techniques inspired from broader machine learning literature as well as focused on our problem domain of CP as a way to increase the amount of data and allowing us to use advanced modeling techniques such as deep neural networks.
- We analyze the impact of data segmentation parameters within our CP context, particularly window length, on automatic detection accuracy. Due to the variability in body movements, the optimal data segmentation parameters are also expected to vary. By means of a thorough analysis, we study the relation between movement type and segmentation length and uncover that a 3-seconds sliding window provides the optimal performances.

- Through a comprehensive range of experiments, we believe our proposed methods establish an important benchmark result for within-movement continuous automatic detection of protective behavior under CP. We conduct experiments within the EmoPain [10] dataset which contains instructed and spontaneous movements captured from people with CLBP (and healthy control participants). The agreement of model and ground truth is comparable to the agreement of human expert judgments.

The paper is structured as follows: Section 2 provides a review on the state of the art of the pain behavior researches. We provide such overview that is outside of the clinic chronic pain rehabilitation to clarify the needs for such system from a wider aspect. Later, previous works on automatic analysis of pain behavior and neural network methods for relevant areas are reviewed in Section 3. We then describe an overview of solutions and in detail the neural network architectures we used in our studies in Section 4, and briefly present the EmoPain dataset [10] used and our data preparation methods in Section 5, while results are reported in Section 6 with a discussion of these results in Section 7. We provide a conclusion of our findings in Section 8.

2 PAIN BEHAVIOR IN CHRONIC PAIN REHABILITATION

The use of body movement as a modality for automatic pain-related detection has been largely ignored even though bodily behaviors such as protective behaviors are more pertinent to pain experiences than facial or vocal expressions [14]. The relevance of the body lies in its indication of action tendency, which in the case of pain is to protect against self-perceived harm or injury [14][15]. The body is an effective modality for automatic detection of affect although most of the work in this area has been focused on the so-called basic affective states (for survey, see: [16][17]). In this section, we provide reviews of the literature on bodily expressions of chronic pain and describe the needs for chronic pain physical rehabilitation as it shifts from the clinic into people’s home and social life.

2.1 Analysis of Bodily Pain Behavior

The existing systematic analysis of protective pain behavior was proposed by [3]. Using trained observers to manually label videos of patients performing specific movements [3][14], they showed that defined protective behaviors were exhibited by people with low back CP and that such analysis is critical to understand how well a person with CP is coping with the condition and engagement in everyday life. Unfortunately, expert visual assessment is expensive and impractical given the prevalence of CP [18][19], limiting observation to clinical settings, which may alter a patient’s behavior [20] and does not reflect abilities (or struggles) in more complex everyday functioning. As such, the need to better understand such behavior in real-life has raised the need to use technology as a way to monitor such behavior [52][53]. However, the approaches used have been limited to coarse behavior, such as studying how far and where a person move with respect to their home using Fitbit and GPS-based technology [8]. The findings from this study showed limited correlations with key affective variables that characterized the ability of the person to self-manage their conditions. The work was critiqued by the literature to further build on the evidence that it is not the quantity of the movement that matters but is the quality and the type of movement (or aspect of movement) that are avoided that provide insights on the ability of the person to cope with and manage their condition [6].

In addition, as physical rehabilitation in chronic conditions transitions from clinician-directed into self-managed (in the form of self-managed movements or functional tasks such as loading the washing machine [8]), visual inspection becomes unfeasible. On the other hand, self-report of pain behaviors [20] in everyday functioning is unreliable as people with CP may not be conscious of their responses to pain or feared situations [8]. More importantly, self-report does not allow for fine-grained measurement, necessary for insight into subjective experiences [3][9] and informing adaptation of movement activity plans or other forms of feedback.
(e.g. just-in-time reminders to breathe deeply to reduce tension). Despite its limitation, the systematic analysis of movement proposed in the above pain literature suggests that protective pain behavior can be automatically detected, and such capability could be embedded in rehabilitation technology to enable it to better support people with CP at home.

In this work, we focus on a set of protective behaviors, namely guarding (stiffness), hesitation, support (bracing), abrupt motion and rubbing (stimulation), which have been particularly highlighted as observable pain behaviors that can provide insight into subjective pain experiences, and so inform intervention [3][14]. First, they are significantly correlated with self-reported pain and fear-related beliefs [3][21]. Further, unlike facial and vocal expressions which primarily communicate, protective behaviors are more reflective of perceived physical demand [14]. Here, limping is not considered as the motion capture data captured during walking contains too much noise. Table 1 provide a more detailed description of protective behaviors referring to [3][10]. However, rather than discriminating between them, we treat them as a unique class that we referred to as protective behavior. The reason is that the number of instances for each behavior would be too limited to investigate the use of neural networks. In addition, the discrimination that matter in providing personalized feedback is that protective behavior has occurred. With a larger dataset, a finer analysis could be used to further specialize the feedback and intervention, but it is outside the scope of this paper.

| Protective Behaviors     | Definition                                                                 |
|--------------------------|---------------------------------------------------------------------------|
| Guarding/Stiffness       | Stiff, interrupted or rigid movement.                                     |
| Hesitation               | Stopping part way through a continuous movement with the movement appearing broken into stages. |
| Support/Bracing          | Position in which a limb supports and maintains an abnormal distribution of weight during a movement which could be done without support. |
| Abrupt Motion            | Any sudden movement extraneous to be intended motion; not a pause as in hesitation. |
| Rubbing/Stimulation      | Massaging touching an affected body part with another body part, or shaking hands or legs. |

### 2.2 The Various Forms of Chronic Pain Self-directed Physical Rehabilitation

How does self-directed rehabilitation take place in people’s life and what are the requirements for personalized support? Although the existing literature in the area is not extensive as yet, qualitative works by Singh et al. (e.g. [6][8]) provide directives on the needs that should be addressed and the different contexts in which it takes place. We do not address all these needs in our paper but we importantly contribute a detection system that could work in all of the activity contexts.

**Dedicated exercise sessions:** These are home-based exercise sessions. Differently from the exercise sessions in the clinic, they do not generally take place in dedicated exercise spaces rather, they are spread within the home space and possibly outside the home [6]. The location of each session is dictated by factors such as time of the day, weather, reason for exercising and room configuration [8]. For example, it was found that people with CP may start the day with some exercising in bed, to activate the body and empower them to get up [15]. Further, people with CP may take advantage of furniture in a room as equipment for exercising, e.g. using the kitchen counter as support while performing ankle exercises [8] [13] and doing stretches in the shower to take advantage of the warm of the water.

**Functioning:** Given the limited physical and emotional resources that people suffering from chronic pain have [33], their priority is functional activity rather than exercising per se. Pain specialist clinicians also emphasize the priority of valued (functional) activities over exercises. As such, physical rehabilitation beyond clinical setting takes place in form of functional movements. People suffering from CP see functional activities such as bending to load the washing machine, stretching to dust the bookshelf, putting away dishes in the cupboard as moments of exercising [8] [56]. If any physical and emotional resource is available, it is dedicated...
to this form of activity first. In fact, they may even reorganize rooms (if they have the emotional capability to do so) so as to increase the challenge that valued activities may require, e.g. putting a favorite book at the top of the shelf so that they have to stretch to take it.

**Function-supporting exercises:** Finally, people with chronic pain may have to interleave functional movements with exercises, such as stretching, as they find this a helpful strategy for completing extensive functional activities. People reported, for example, to place a mat closing to the dishwasher to interleave its loading operation with stretches of the trunk to relax the muscle while also be able to complete the loading tasks. Indeed, they learn (in pain management clinical sessions) to take advantages, even outside the home, of opportunities for impromptu exercises, e.g. taking a book from a high shelf in a bookstore to perform a stretching exercise [8].

Differently from clinical settings, the above scenarios make it clear that self-directed physical rehabilitation is ubiquitous and as such, technology that aims to support it needs to address this requirement. Previous studies such as [13] discuss the possibility of using wearable sensors to track pain-related body movement behaviors and muscle activity patterns in ubiquitous settings. To provide psychological support in clinical settings, physiotherapists analyze various aspects of a patient’s exercise and tailor suggestions, reminders, and feedback accordingly [1]. For example, when observing the patient using the hands to push up during standing up from their chair (to avoid trunk flexion due to pain-related anxiety), the physiotherapist may suggest that the patient exercise using a taller chair to help him/her become more comfortable while avoiding the use of support to sit and then slowly decrease the height of the seat. The physiotherapist may also provide verbal reward at the same movement phase when psychological progress is observed. This type of movement-phases tailored feedback makes it critical to recognize pain/affect-related behaviors continuously within each movement type.

In this paper, we address the above challenges with our investigation of continuous detection of protective behavior within each instant type of movement considering also its initial merging into subsequent movement. We take advantage of the existing EmoPain dataset [10] as the participants were not constrained in doing movement according to gold standard (typical in exercise setting) but asked to perform sequence of functional-related movements so that the ending of one movement may exhibit transition into another movement.

3 PREVIOUS 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 [22], [23], and [21] on lower back, neck, and knee CP respectively. Dickey et al. [24] and Olugbade et al. [11][12] further discriminate levels of self-reported pain within people with low back CP. A common finding in these studies is that the way a person with CP uses (or avoids the use of) a painful anatomical segment provides information about subjective experiences. [13] investigated movement behaviors that clinicians use in judging pain-related self-efficacy and showed the feasibility of automatic detection based on these cues. Interestingly, protective behaviors were the cues of low self-efficacy specified by clinicians, and the authors used features based on the method of [18] on automatic detection of protective behavior to characterize it. [13] further provides evidence that low-cost body sensing technology can enable the detection of pain related experiences in functional movements.

More directly relevant to our work is [10] where Aung et al. present the EmoPain dataset (also used in [11][12][13]) which includes motion capture and sEMG data recorded while people with CLBP (and healthy control participants) performed 6 everyday movements typically challenging for this cohort. The authors used the range of angles for 13 full-body joints, the mean energy for these joints and the mean sEMG recorded bilaterally from the lower and upper back muscles for each instance containing multiple movements. These were used to predict the mean (across 4 pain-expert raters: two physiotherapists and two clinical psychologists) of the proportion of the instance that had been labelled as protective based on Random Forests (RF). They obtained between 0.019 and 0.034 mean squared error (mean = 0.027, standard deviation = 0.005), however, Pearson’s correlation was between 0.16 and 0.71 (mean = 0.44, standard deviation = 0.16). The low correlation despite low error suggests that although the predicted values were close to the ground truth, these errors are
not consistent in their direction (positive versus negative). Previous classification of a subset of these data focusing on two types of the movement achieved better results of 0.81 and 0.73 F1 score respectively [25].

One important limitation of the above studies is that separate models were built for the different types of movement. In addition, detection was only per overall movement instance rather than a more fine-grained level as necessary for providing more personalized support. In this paper, we build on these studies by investigating within-movement continuous detection based on one classification model for all five movement types (sit-to-stand, stand-to-sit, standing on one leg, reaching forward and bending). We also extend this work by including the modeling of bracing and stimulation behavior available in the data set but not addressed in [10]. To address the temporal aspect of the movement, our work is based on neural networks LSTM layers; the LSTM layer is particularly designed for sequence data such as is available in the EmoPain dataset.

3.1 Neural Network Models for Behavior Detection

Neural networks, particularly deep networks, is currently the leading approach in many previously very challenging machine learning tasks such as image recognition, with increasing use in healthcare [26]. As far as we know, the only studies using this method in the area of automatic detection of pain behavior have focused on detection from facial expressions. Much of these has been facilitated by the publicly available UNBC-McMaster database [27], which contains about 200 sequences of over 40,000 face image frames collected from 25 people with clinical (shoulder) pain [28] during a variety of physiotherapist-guided movements. To name a few, [29] used a stack of 3 LSTM layers on image sequences of size 16 (in frames) extracted from the original data and pre-processed by applying super resolution after down-sampling on each image. The authors achieved accuracies of 0.56, 0.76, and 0.39 for 3 pain groups based on the 16-level pain scale. In [30], the original data was split into RGB channels, the image at each frame was then flattened into a vector and length-fixed sequences were randomly selected to obtain the training set. For the test set, a sliding-window of the same length was passed through the held-out data and zero paddings were added for sequences shorter than the window length. Their results show a mean squared error of 1.54 with Pearson’s correlation of 0.65, in discriminating between the original 16 levels of the pain scale. [31] improved on this performance by using a deep network previously trained for face identification but replacing its fully connected layers with 2 untrained layers having fewer units and a dropout layer in between. Their performance was 0.80 of mean squared error. [32] achieved better performance similarly using a pre-trained network. They replaced the last fully connected layer of a convolutional network with an LSTM layer and used it on image sequences pre-processed by removing consecutive frames of the majority no-pain label from each video instance until it matched the collective number of pain frames in the instances; 50% of the images were then flipped with random noise applied to facial landmarks. The performance they obtained was 0.74 mean squared error with 0.78 Pearson’s correlation for the 16-level scale and 0.93 accuracy for pain versus no pain classification. Although it is difficult to directly translate the outcomes of these studies to work using motion capture and sEMG data, their findings suggest that the recurrent learning performed by LSTM layers contribute to automatic detection of pain behavior in sequential data.

Findings in human activity recognition further point to the efficacy of convolutional and LSTM networks with body movement data. For example, [33] used a bidirectional LSTM to classify physical activities in the Opportunity [34] and PAMAP2 [35] datasets. They obtained mean F1 scores of 0.75 and 0.94 on the two datasets respectively using hold-out validation. In this study, data samples were frames of lengths of 1 and 5.12 second(s), with overlapping ratio of 50% and 78% respectively, from the movement instances. [36] 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-layers LSTM networks with dropouts after each layer. This method further led to mean F1 score of 0.92 on the Skoda dataset [37]. Particularly, they proposed to train the model with data segmented with multiple window lengths in a bootstrapping manner while the inference was conducted directly on each single timesteps/samples. [38] used a stack of two convolutional followed by max pooling, one (more) convolutional, LSTM, and dense (with softmax activation) layers trained on the Opportunity dataset to classify the activities in the Skoda dataset. [33] further used a three-layer LSTM network to automatically detect freezing behavior in 10 people with
Parkinson’s disease while they performed everyday activities, using data from the Daphnet Gait [39] dataset. Based on motion capture data from around the ankle, knee, and trunk, they obtained mean F1 score of 0.76 with hold-out validation. Given the similarity of the problem we address in this paper and theirs, we focus on a three-layer LSTM network for automatic detection of protective behavior although we will compare its performance with the convolutional LSTM network used in [38].

There are few other studies where the detection of anomalous movement behaviors (such as due to a medical condition) have been investigated. Such tasks are more challenging as these behaviors are embedded, as modulations [40], in the performance of physical activity. One of these works is from Rad et al. [41] who used a network of 3 convolutional layers, each followed by an average pooling layer, on motion capture data in the Stereotypical Motor Movements (SMMs) [42][43] dataset recorded from the wrists and chest of 6 people with autism spectrum disorder. Their goal was to detect stereotypical movements within window lengths of 1 second (overlapping ratio of 87%). The SMMs dataset contains two streams of data with one stream collected in the lab and the other in classrooms, and their result of mean F1 score of 0.74 with the lab data outperformed the traditional feature engineering method with Support Vector Machines and RF used in [42][43]. Unsurprisingly, the average F1 score obtained was only around 0.5 with the classroom data, where movement is less constrained and noisier although the poorer performance may also be due to smaller volume of data.

Beyond the greater challenge of detecting anomalous movement behaviors (compared to the recognition of physical activity types) in data from real patients, this area also faces the difficulty of obtaining large volume of training data of the positive class(es), leading to considerable skew in the datasets that exist, and also constraining the use of deep neural network models. Although LSTM networks show a lot of promise based on our review, care must be taken for how the input data is formatted, particularly in the approach taken to segment the data in the temporal dimension. Previous works, such as the studies discussed above, have employed window-based segmentation, where most training instances are fixed-length frames generated from the dataset used. This method is suitable for real-time applications because it enables detection in small-continuous streams of data through time and so we use this approach in this paper. As far as we know, except for [36] that used dynamic window lengths to generate training data, there has previously been little discussion or justification for choices of window parameters, such as the length of the window, that used in our approach even though these are strongly related to system performance [46]. We address this problem in this paper. To also support our discussion on window parameters, the idea raised from [36] about training with dynamic frames and inferencing on single timesteps will be tested in our scenario.

4 SOLUTION OVERVIEW

In this section, we first define the research scope by giving several considerations and respective solutions. Then, we describe in detail the neural network architectures.

4.1 Design Considerations

Toward using neural networks for protective behavior detection, our research scope is defined by following considerations:

- **Modeling Temporal Nature.** Given that motion capture data and sEMG data are typically formatted in temporal sequences and that the volume of labelled data (e.g. the EmoPain dataset used in this study) is quite limited, a less deep RNN architecture is proposed to detect protective behavior.

- **Emphasis on Within-Movement Continuous Detection.** As protective behavior is exhibited along with the execution of specific movement/activity, and physiotherapists make the judgement based on patients’ performance during movement, our work is aimed to automatically detect such behavior within instances of movements. For future real-life application, an ideal system would conduct a 2-stages process, namely the HAR stage for recognizing the movement of interest and protective-behavior-detection stage given such detected movement period. In this work, we focus on the second stage, i.e. detection of protective behavior once a movement of interest has been identified. Our data segmentation pipeline presented in the next section would reflect this idea.
- **Limited Dataset.** To make the available data suitable for the development of state-of-the-art techniques, we will conduct data augmentation with 3 different approaches, namely Reversing, Gaussian Noise and Random Discarding. These approaches would be discussed in detail in next section.

### 4.2 Protect-LSTM: Network Architecture

Unlike the convolutional neural network (CNN), which is powerful for extracting spatial information, recurrent neural networks (RNNs) have shown good capability for learning from time-dependent data streams in speech recognition and natural language processing. Given inherent dynamic nature of motion capture and sEMG data, we use RNN components to build our network. A typical forward RNN structure that connects in forward time is shown in Fig. 1 (a), with the input as a temporal sequence and computed state information passed forward along the network.

![A forward RNN structure](a)  

**Fig. 1.** (a) A forward RNN structure. (b) A traditional LSTM unit. (c) The applied forward Protect-LSTM network.

At the core of any RNN architecture is the processing unit indicated by $A$ in Fig. 1 (a). One of the most widely applied unit in RNNs is the LSTM [47] which solved the vanishing gradient problem which traditional RNNs faced in backpropagation over a long temporal range. Every LSTM unit updates its internal state based on previous information, and it is able to store temporal information [47]. To extract long-term temporal information in a direction natural to the expression of protective behavior in physical movements, we focus on forward information pass in our architecture. As precise timing is not an important factor and a greater concern, in order to simplify the network to avoid overfitting, the LSTM unit that we use in this work is the *vanilla* variant without peephole connection [48] which can be seen in Fig. 1 (b).

The Input of a LSTM unit is the current input data $X_t$, previous hidden state $h_{t-1}$ and the previous cell state $C_{t-1}$, while the output is the current hidden state $h_t$ and cell state $C_t$. Unlike the works on HAR that the input data at current timestep usually concatenates a certain length of time to make a single vector. Here, the input data $X_t$ equal to a single sample in the frame at timestep $t$. By using this strategy, the output of at each timestep is based on the previously consecutive knowledge acquired. The states are updated with several computation
gates like Input Gate with output $i_t$, Forget Gate with output $f_t$, Output Gate with output $o_t$ and Cell Gate with output $\tilde{c}_t$. The computation within a LSTM unit at timestep $t$ which in our case represents a single frame in the wearable data sequence is given as below:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$  \hspace{1cm} (1)

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$  \hspace{1cm} (2)

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$  \hspace{1cm} (3)

$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$  \hspace{1cm} (4)

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$  \hspace{1cm} (5)

$$h_t = o_t \odot \tanh(c_t)$$  \hspace{1cm} (6)

and

$$\sigma(x) = (1 + e^{-x})^{-1}$$  \hspace{1cm} (7)

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$  \hspace{1cm} (8)

where $\odot$ denotes element-wise multiplication, $W_{\cdot}$ are weight matrices and $b_{\cdot}$ are bias vectors which are used to represent the transformation of each gate. The processing at timestep $t + 1$ would take the current output $c_t$ and $h_t$ to iterate the same computation mentioned above. To also use the experience provided by [33] [36] that LSTM networks outperform other network architectures like CNN on processing temporal sequences collected with wearable motion capture sensors, while also the smaller dataset used is unsuitable for convolutional layers and deeper models, the network we eventually proposed is Protect-LSTM with three LSTM layers computing on a single forward direction as shown in Fig. 1 (c). The configuration of Protect-LSTM is described in the next subsection, while the number of layers and hidden units would be further justified in Section 6.2. As we would examine the parameter impact of the sliding-window, the length of the input layer is adjusted to the length of the input data frame created by different windows respectively.

For our work, the last output hidden state $h_{t+N}$ would be used as input for a fully-connected layer with softmax activation for classification. For the current input frame $X_T = [x_t, x_{t+1}, ..., x_{t+N}]$ (generated by sliding-window segmentation with length of $N + 1$), given all the output hidden state $h_T = [h_t, h_{t+1}, ..., h_{t+N}]$ from the last LSTM layer the computation of class probability $P = [p_1, ..., p_K]$ where $K$ denotes the number of classes (in our case $K = 2$), and the final label prediction $Y$ can be written as follows:

$$P = \text{softmax}(W_{hk}^T h_{t+N} + b_k)$$ \hspace{1cm} (9)

$$Y = \underset{k}{\text{argmax}}(p_k)$$ \hspace{1cm} (10)

where $W_{hk}$ and $b_k$ are weight matrix and bias vector used in the fully-connected layer.

On the other hand, we would also do a sample-wise prediction in Section 6.3.3 following [36], where each output state $h_t$ would be used as input for a fully-connected layer with softmax activation for classification. For the current single timestep $t$, given similar input and output of the last LSTM layer as above, the computation of class probability $P_t = [p_{t1}, ..., p_{tK}]$ and the label prediction $Y_t$ can be written as follows:

$$P_t = \text{softmax}(W_{hk}^T h_t + b_k)$$ \hspace{1cm} (11)

$$Y_t = \underset{k}{\text{argmax}}(p_k)$$ \hspace{1cm} (12)
Additionally, based on Protect-LSTM, we explore the use of an architecture that processes motion capture data and sEMG data separately, called Dual-stream LSTM. As shown in Fig. 2, each stream of the network is a three-layers LSTM network (Protect-LSTM) while representational layer fusion is conducted at decision level.

Fig. 2. The Dual-stream LSTM networks where motion capture and sEMG data are separately processed.

4.2.1 Model Configuration. One of the important factors in choosing a network architecture is to decide on the number of layers in the network and number of hidden units in each layer. To this end, we perform a series of experiments (details in Section 6.2) to find the optimal hyperparameters for our target task and dataset. Our results show that for Protect-LSTM, three LSTM layers each containing 32 hidden units and followed by a Dropout layer with probability of 0.5 provide the best accuracy on the task of detecting protective behavior, as such we use this configuration in rest of the paper. As it is a binary classification task, the number of the output node in the last softmax layer is 2. Similarly, for the Dual-stream LSTM, the number of hidden units in motion-capture stream and sEMG stream is 24 and 8 respectively, where each LSTM layer is also followed by a Dropout layer with probability of 0.5. The weights for loss-updating applied to both streams are equal.

For comparison purposes, all the neural network methods used in our experiments employed the Adam [49] to update the weight, and the learning rate is fixed to 0.001. The mini-batch size is determined according to the size of the training set. For all the neural network methods, as the sizes of training sets are similar: the mini-batch size is fixed to 20. The deep learning framework is implemented using TensorFlow plus Keras. The hardware used is a workstation with Intel i7 8700K CPU and the GPU acceleration is not employed, while the average training time of Protect-LSTM using the Emo-Pain dataset after augmentation is around 10ms per iteration.

5 DATASET AND METHOD

In this section, we first present the EmoPain dataset that we used to model protective behavior. Then we discuss our data pre-processing and data augmentation pipelines, followed by a description of our validation methods.

5.1 The EmoPain Dataset

The Emo-Pain dataset [10] contains motion capture and sEMG data collected from 26 healthy people and 22 CLBP patients doing specially selected physical movements. Healthy participants are included as they represent different ways of moving with their idiosyncrasies and differences in fitness levels, rather than considering a standard gold model of movement execution. Despite the four expert raters (2 physiotherapists and 2 clinical psychologists) have checked the movements performed by healthy subjects as well, no protective labelling was assigned to any of them. Whilst the original dataset contains data from 22 patients, 4 patients were left out because of errors in their sEMG data recordings. In order to avoid biasing the model towards healthy participants, 12 healthy people were randomly selected. This is because healthy participants may have more execution of the same movement in comparison to the CLBP participants. As a result, the data used is collected from 12 healthy people and 18 CLBP patients.

Fig. 3 show examples of protective and non-protective behaviors from the EmoPain dataset. These avatars or stick figures were built directly from participants’ motion capture data and represent instances of movements from the dataset. For visualization purpose, for each sequence, we show here only representative moments of the movement. Hence, the length of each sequences is not representative of the real movement duration and should not be compared across sequences. The average upper envelope of the rectified sEMG data collected from two places on the lower back is also provided for each avatar sequence respectively.

The gold standard reference of a movement does not reflect the way people are expected to move as physical...
rehabilitation is moving away from standard movements towards the main target of ability to function. Fig. 3 (a) shows differences in the performance of a reaching forward movement between a healthy participant at the top and two CLBP patients below. We can see the differences in stretching ranges and also the different strategies, with the latter simply raising the arms but not bending forwards. We can also observe the different strategies with the bottom patient keeping the feet closer together making bending more difficult. Often people with CP are unaware of avoiding-facilitative movements as their attention is on pain rather than proprioceptive feedback. We can also notice that the middle patient remains in guarded position (not relaxed) at the end of the movement for fear of increased pain. Similarly, such protective strategies can be observed in the CLBP participant performing a stand-to-sit movement in Fig. 3 (b). Differently from the top healthy participant, the stick figure does not bend the trunk but exploits 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. These are just examples of strategies used by people with CP as each person personalize the strategies to their physical capabilities and their own understanding of what could be a dangerous movement.

The five specific movements we chose to extract are bending, standing on one leg, sit to stand, stand to sit and reaching forward performed by the 30 subjects under or without instructions, which covered almost all the movements in the dataset while the rest represents transition movements like standing still, sitting still and walking around. These movements aim to train the physical and psychological capabilities needed for everyday function such as reaching forward to take an object from a shelf or across a table or bending to load the washing machine.

Two trials with different difficulty levels of the five movements are applied to some participants including both healthy and CLBP participants. During the normal trial, participants were free to perform the movement as they pleased, e.g. they could stand on their preferred leg, they could start the movement at any time they wished. For the difficult trial, participants were asked to start on a prompt from the experimenter, and to carry a 2Kg weight in each hand during reaching forward and bending. These more difficult versions of the same movements simulated situations where a person is under social pressure to move or is carrying bags. Again,
these more difficult versions are often suggested by physiotherapists to gain confidence in moving outside the home [14]. As a result, we treat two trials of movements performed by one subject as two different instances. 5 healthy people and 11 CLBP patients did movements at both levels of difficulty.

Therefore, we have 17 instances (5×2+7) from healthy people and 29 instances (11×2+7) from CLBP patients, which make 46 instances in total, where each instance contains all the selected movements performed by one participant at one level of difficulty. More details about the dataset is provided in [10].

5.2 Data Preparation

In this section, we describe the data processing pipeline we apply on the EmoPain dataset to enable us to model protective behavior.

To avoid ambiguity, we clarify that an 'instance' is referred to the data sequence containing all the movements performed by a subject during one trial; 'frame' is a small segment containing several samples within a data instance; 'sample' is a single data vector at each single timestep (for our case is at 1/60 second as the sensor was operating at 60Hz).

5.2.1 Low-level Feature Computation. In the EmoPain dataset, the motion capture data is organized as temporal sequence of 3D coordinates along with the velocity values collected from 26 microelectromechanical (MEMS) based IMUs at 60Hz. We computed 13 low-level features suggested in Aung et al [10] corresponding to 13 inner angles in 3D space based on the 26 anatomical points. Also, we computed 13 'energy' features using the sum square of the angular velocities at each angle. The muscle activity is represented with the upper envelope of the rectified sEMG data collected from four places on the back. We therefore compute 30 features in total from each sample, including 13 inner angles, 13 energies and 4 upper envelopes of the rectified sEMG data from the original dataset to create a data matrix. To maintain the temporal order of the data, the data matrix is organized as Fig. 4. As we can see, the dimension of matrix (the column) is used to fill the 30-dimensional data at each timestep or sample, while the length of matrix (the row) is used to put the data in a temporal order.

![Fig. 4. The data matrix of an instance. A1 to A13 are the inner angles, E1 to E13 are the energies and sEMG1 to sEMG4 are the rectified sEMG data.](image)

5.2.2 Data Segmentation. Both for the training and testing set, a sliding-window segmentation method [57] is applied to generate frames of continuous portions of data instance. The parameters related to the sliding-window are justified and analyzed on the basis of the different movement types in a later section.

Fig. 5 shows a data instance, i.e., a sequence of movement timesteps done by a subject during one trial. It consists of several movement types (e.g., bending and standing on one leg) which are separated by transition movements such as standing or sitting still. As explained earlier in Section 4.1, a key design consideration for our work is to support detection of protective behavior when a participant is engaged in a particular movement. For example, we envision that during daily physical rehabilitation the user would perform a particular movement such as bending, and our model will automatically analyse the raw data to detect protective behavior. As such, we are only interested in modeling protective behavior within a movement to detect when protective events occur as opposed to an open-ended detection. Therefore, we segment motion capture and sEMG data into frames containing a moment of movement. That said, our model does not take the type of movement as an input in the training process, but instead aims at generalizing the automatic protective behavior detection across all types of movements. The long-term aim is to develop a classification model that is movement independent and can be easily applied to new type of movements.
Another critical issue in data preparation is to handle edge cases when the sliding window is at the end of a movement instance. We explore three different ways of handling this scenario:

a) 0-padding: We pad the frame with zeroes. This is a typical approach used in activity recognition based on computer vision literature and rationale [59] [60].

b) Last-padding: We use the last sample of that movement and repeatedly add it to the frame. A clear downside of this approach is that by repeating the last sample several times, it could be interpreted as a ‘guarding’ or ‘stiffness’ behavior by the model.

c) Next-padding: We use the following samples for padding, as a way to simulate continuous natural transitions between movements.

A comparison between these padding methods will be conducted in Section 6.2. By using a sliding-window with length of 3 seconds and overlapping ratio of 75%, the total number of frames generated from 30 subjects is 2569.

5.2.3 Ground Truth Computation. According to [10], the labelling of protective behaviors was completed separately by four expert raters, 2 physiotherapists and 2 psychologists with clinical experience in chronic pain. Each expert rater inspected every patient’s video (gathered in synchrony with the motion capture data and sEMG data) and marked the data samples where the protective behavior started and ended. In our case, the sEMG could enable the system to have better insights on how the movement is performed. However, our aim is not to evaluate the physiotherapists but to understand if the system would be able to reach performances that are at least as good as the one of the physiotherapists. Fig. 6 presents a visualization of the coding result in one of the movement instances of one patient and the level of agreement between raters for that movement.

![Fig. 5. The applied sliding-window segmentation and padding. W is the window length, S is the sliding step.](image)

![Fig. 6. The visualization of the binary coding for protective behavior by 4 raters of an instance.](image)

Similar to findings in other related studies [16] [61] and as discussed in [10], ratings of non-acted behaviors from different observers do not perfectly agree. It can also be seen that such behavior is not necessarily continuously expressed through the movement (for a review of similar findings in other studies, see [62]). This suggests that pain-related expressions occur in parts of the movement which the patient fears, or experiences increased pain.

Following a typical approach for building the ground truth for affective computing [16], we used a sliding-window to segment movement instances and defined the label of a frame based on a majority-voting manner: the frame is labelled protective if at least 50% of the samples within it were considered protective by at least two
A sample within a frame is considered protective if it is included in the protective period marked by at least two raters.

The rationale behind frame level approach is that the label of a frame needs to capture the relevant (affective) need within that frame, rather than merely mathematically encapsulate the labels of the samples within the frame. From the modeling perspective, a system should be trained to detect the salient moments of affective states within a frame rather than to learn from artificial and pre-segmented positive/negative samples.

5.3 Data Augmentation

To lessen the overfitting risk when applying neural networks on comparatively smaller datasets for CP, e.g. the EmoPain database, we propose to use three different data augmentation methods designed to mimic real-life situations:

i) **Reversing**, which is to reuse the data with a temporally reversed direction. This method is proposed as the movement included in EmoPain dataset is cyclic, e.g. stand to sit and sit to stand;

ii) **Gaussian Noise**, which is to simulate the signal noise that may exist during data capturing in real life. We create the normal gaussian noise with three standard deviations of 0.05, 0.1, 0.15 and globally add them to the original data respectively;

iii) **Random Discarding**, which is to simulate unexpected data lost at some points. We randomly set the data at some timesteps as well as at some body parts to 0 with selection possibilities of 5%, 10% and 15% respectively.

Note, all these three methods would not change the temporal order of the data as well as the movements to a noticeable degree. Therefore, the labels stay unchanged. After testing those three data augmentation methods separately with Protect-LSTM on the entire experiment dataset, we found that only the last two methods (gaussian noise and random discarding) are able to improve performance. Consequently, we combine those two approaches as the default augmentation method for the following experiments. The number of frames after using such combined augmentation method is 18653. A detailed comparison among the three augmentation methods is reported in Section 6.2.

5.4 Validation Methods and Metrics

Three different validation methods are used to evaluate the performance of models. The original hold-out validation is to leave the data of some subjects for testing and the rest for training. Here, we conduct the 6-folds-hold-out validation, where at each fold the data of 5 out of the 30 subjects are manually left out and used for testing. This ensures that the model is tested for generalization over all individual participants. To balance the number of CP and healthy participants, we ensured that each test fold would contain data from 3 CP and 2 control subjects respectively. We call such validation a cross hold-out validation. As we envision that the use of our model will be in the context of personal rehabilitation where the model can be further tailored to the same individual, a cross validation by leaving some instances out (LSIO) is also used, where data (but not the same instances) from a same participant appear both in training and test set. Further, the standard Leave-One-Subject-Out (LOSO) validation is applied as well to further demonstrate the generalization capabilities of the modelling approach towards different individuals.

5.4.1 Validation Metrics. Given the protective behavior detection is a binary classification problem in our scenario where the detection of both protective and non-protective behavior are similarly important, we report the mean F1-Score as a metric that ignores the volume imbalance between classes. Furthermore, such metric is in line with other works [33] in relevant area. The mean F1-Score \( F_{m} \) is computed as follow:

\[
F_{m} = \frac{2}{|c|} \sum_{c} \frac{pre_{c} \cdot recall_{c}}{pre_{c} + recall_{c}}
\]

where \( pre_{c} \) and \( recall_{c} \) is the precision and recall ratio of class \( c = 2 \) classes (protective and non-protective). Moreover, for completeness, the accuracy, mean precision, mean recall and confusion matrices are reported. To further understand how different architectures and parameters compare with each others, we carry out
statistical tests (repeated measures ANOVA and post-hoc paired t-tests) on the LOSO evaluation results. For other evaluation approaches, a thorough statistical analysis was not possible given the limited number of folds.

5.5 Comparison algorithms

For comparison, we use CNN network, bi-directional LSTM network (bi-LSTM) and Convolutional LSTM network (Conv-LSTM) mentioned in [41] [33] [38] to show the advantage of using LSTM and in particular Protect-LSTM. In addition, we consider method with RF as a traditional one based on feature engineering as it was used in [25] to model guarding behavior (one category of protective behavior) in the EmoPain dataset. It should be noted that differently from [25], we aim to perform the modeling across different movement types rather than for each movement type separately to move towards real life situation where the movement type is not known in advance. The cross hold-out, LSIO and LOSO validations mentioned in Section 5.4 are used. In the case of the LOSO validation (given the sufficient number of folds ~ 30), repeated measures ANOVA and post-hoc paired-tests with Bonferroni corrections are used to understand to what extent the differences in performances between the various approaches are statistically significant.

For all the neural network methods, a sliding-window segmentation is used with window length of 3s and overlapping ratio of 75%, while Adam [49] is used for weight updating. For the RF model, traditional features are extracted from the 3s frames. The hyperparameters of each method were chosen through an optimization analysis. Further details about each comparison algorithms are provided below:

**CNN [41].** The 3-layers CNN architecture used in this work is implemented according to [41], while the classification result is produced by a softmax layer at final stage instead of using an extra SVM classifier. The convolution kernel size is 1 × 10, max pooling size is 1 × 2 and number of feature maps is 10. The mini-batch size is set to 20.

**ConvLSTM [38].** The ConvLSTM was previously used on larger databases for HAR [38] with wearable motion sensors, while better results were achieved than using normal LSTM networks. Here we also use this model to conduct a comparison experiment to see if the introduction of convolution is able to improve the detection accuracy on our task. The architecture is the same that was used in [38]. The size of the convolution kernel is set to 1 × 10, while max pooling size is 1 × 2 and the number of feature maps in convolutional layers and hidden units in LSTM layers is set to 10 and 32 respectively. The mini-batch size is set to 50.

**bi-LSTM [33].** As an alternative flavor of LSTM network, bi-LSTM utilize context information in the ‘past’ and the ‘future’ to compute the output at each timestep. We implemented the bi-LSTM according to [33]. Through grid search, the hidden units in each LSTM layer is set to 16 with mini-batch size set to 20.

**Random Forest (Traditional Feature Engineering Method [25] used on EmoPain dataset).** We use a RF algorithm with 30 trees for frame-based detection. We call it RF-frame. First, we extract length-fixed feature vectors for each frame, with a total number of feature vectors computed after augmentation being 18180. Those feature vectors are further divided into training-testing pairs based on the cross hold-out, LSIO and LO SO validations. The feature computed comprises the range of the joint angles, the means of joint acceleration value and the means of rectified SEMG value, which were used in [25]. For the data matrix described in the previous subsection, the dimension of each feature vector extracted is 30.

6 RESULTS

In this section we first present the results achieved with Protect-LSTM, Dual-stream LSTM and the comparison algorithms. After that, we report the justification of model configuration, results achieved with three different padding methods as well as the three proposed augmentation methods. Finally, we report the experiment conducted to understand the impact of window length on protective behavior detection by analyzing the different movements separately first and together following up.

6.1 Automatic Detection of Protective Behavior
The results obtained with each algorithm are given in Table 2. We can see that the Protect-LSTM achieves a best mean F1-score of 0.815, 0.742 in LOSO and LSIO validations respectively while Dual-stream LSTM achieves a best mean F1-score of 0.742 in cross hold-out validation.

Table 2. Results using CNN, Conv-LSTM, bi-LSTM, Dual-stream LSTM and Protect-LSTM with window length, overlapping ratio = 3s, 75% and traditional feature extraction method with RF on frames respectively. The validation methods are Cross Hold-out, Leave-one-subject-out (LOSO) and Leave-some-instances-out (LSIO) validations.

|                | RF-Frames | CNN  | ConvLSTM | bi-LSTM | Dual-stream LSTM | Protect-LSTM |
|----------------|-----------|------|----------|---------|-----------------|--------------|
| **Cross Hold-out** |           |      |          |         |                 |              |
| Accuracy       | 0.6248    | 0.6289 | 0.6166  | 0.7084  | **0.7478**      | 0.7436       |
| Fm             | 0.55      | 0.542  | 0.607   | 0.693   | **0.742**       | 0.735        |
| Recall         | 0.5650    | 0.5624 | 0.6107  | 0.6918  | **0.7476**      | 0.7377       |
| Precision      | 0.5986    | 0.5936 | 0.6072  | 0.6950  | **0.7397**      | 0.7326       |
| **LOSO**       |           |      |          |         |                 |              |
| Accuracy       | 0.7207    | 0.7736 | 0.7913  | 0.8033  | 0.8035          | **0.8686**   |
| Fm             | 0.67      | 0.697  | 0.767   | 0.794   | 0.795           | **0.815**    |
| Recall         | 0.6674    | 0.6923 | 0.7565  | 0.7933  | 0.7956          | **0.8108**   |
| Precision      | 0.7366    | 0.8045 | 0.7959  | 0.7954  | 0.7938          | **0.8195**   |
| p-value (<0.05)| 0.004     | 0.003  | 0.032   | >0.05   | >0.05           | -             |
| **LSIO**       |           |      |          |         |                 |              |
| Accuracy       | 0.593     | 0.6703 | 0.655   | 0.7336  | 0.7256          | **0.7485**   |
| Fm             | 0.544     | 0.605  | 0.653   | 0.72    | 0.718           | **0.742**    |
| Recall         | 0.5476    | 0.6111 | 0.67    | 0.7287  | 0.7235          | **0.75**     |
| Precision      | 0.5566    | 0.6653 | 0.6631  | 0.7228  | 0.7161          | **0.7406**   |

A repeated measures ANOVA showed significant difference in performance (LOSO mean F1 scores) between the algorithms: F(0.651, 4.054)=6.311, p<0.001, $\mu^2$=0.179. Further post-hoc paired t-tests with Bonferroni correction (see Table 2) shows that the Protect-LSTM performs significantly better than the RF-frames (p=0.004) and CNN (p=0.003). It also shows that bi-LSTM is not significantly different from Protect-LSTM (at significance level p=0.05) but is better than RF-Frames with close significance (p=0.061). The Dual-LSTM and Conv-LSTM do not significantly differ in performance with any of the other methods. This suggests that Protect-LSTM does indeed provide overall better performance. These results support previous findings [33] [36] that recurrent models like LSTM network are better at processing time sequence data. Interestingly, the Conv-LSTM performs slightly better than the CNN, possibly because it is designed to integrate temporal information in such forms of data.

For the 18 folds in LOSO validation where testing subjects are patients, we further computed two-way mixed, absolute agreement intraclass correlations (ICCs) to compare the level of agreement between the ground truth (based on labels from the expert raters) and the Protect-LSTM with the level of agreement between the expert raters. The ICC is a standard method for computing interrater agreement [45]. The absolute agreement ICC, which we used, measures strict agreement, rather than the more liberal similarity between rank order of the alternative ‘consensus’ ICC [58]. A two-way mixed model was used in order to account for rater effect [58]. We found ICC = 0.215 (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 Protect-LSTM and the ground truth based on the labels from these raters. This finding suggests that Protect-LSTM is able to provide excellent level of agreement [44] with the average expert rater, which aligns with the goal of our modelling. The agreement is also higher than that between the raters although this may be explained by the fact that unlike the raters, whose ratings are based on their independent experiences and background (even if they did have discussions to resolve rating disagreements), the algorithm’s training is solely based on the average rater’s labelling.

The confusion matrix for the result achieved with Protect-LSTM using cross hold-out validation is given in Table 3. We can see from the table that, as the model was also running on healthy subjects, the protective behavior has been detected at some points. In particular, after checking with previous labelers as well as the videos and the data animations of several specific healthy subjects, we identified various reasons for possible misclassification: i) some healthy participants were not familiar with the movement or instructions from experimenter so hesitated when performing it; ii) some were not able to conduct specific movements normally...
Automatic Detection of Protective Behavior in Chronic Pain Physical Rehabilitation: A Recurrent Neural Network Approach • XX:17

Table 3. Confusion matrix for the detection performance of Protect-LSTM using the window length of 3s in cross hold-out validation

| Ground truth | Non-protective | Protective |
|--------------|---------------|------------|
| Non-protective | 1225 (76.56%) | 375 (23.44%) |
| Protective    | 301 (29.03%)  | 736 (70.97%) |

like reaching forward due to other physical issues like obesity instead of CP.

6.2 Justification of Modelling Configuration

In this subsection, we first justify the configuration of Protect-LSTM about number of layers and hidden units. Then we report the performances achieved with three different padding methods applied during data segmentation, namely to pad the end of the movement (both mocap and sEMG data) with 0 (0-padding), with repeating the last sample (Last-padding) and with the sample following up (Next-padding). In the end, the three augmentation methods proposed in Section 5.3 are also compared.  

6.2.1 Configuration of Protect-LSTM. For experiments conducted above, we used Protect-LSTM with 3 layers and 32 hidden units in each. Here we justify such a parameter setting through a cross hold-out validation on the entire dataset. When comparing the number of layers, the number of hidden units in each layer is set to 32; the number of layers is set to 3 when comparing the number of hidden units. Results are given in fig. 7, which support our previous configurations of using a 3-layers LSTM network with 32 hidden units at each. However, such hyperparameter setting is sensitive to different databases.

![Fig. 7. Justification of the configuration of Protect-LSTM.](image)

6.2.2 Padding Methods. In Section 5.2, we used the sliding-window for data segmentation among different movements where the frame was padded with 0 (0-padding) if the window slides across the end of one movement. As there are other two ways of padding by repeating the last sample of that movement (Last-padding) or using samples following up that movement (Next-padding), here we conduct a comparison experiment among these three methods. The Protect-LSTM is used separately with the three padding methods on the whole dataset under three validation methods (LOSO, cross hold-out and LSIO validations). The hidden unit of each LSTM is set to 32, mini batch size is fixed to 20, the drop-out ratio at each layer is set to 0.5 while the weight is updated using the Adam algorithm [49].

A repeated measures ANOVA is carried out to understand if the differences of performances (LOSO mean F1 scores) among the three padding methods are statistically significant. Given that sphericity could not be assumed (p<.001), Greenhouse-Geiser correction was applied to the degrees of freedom. Results are summarized in Table 4. The results show an effect of padding method on performances (F(1.265, 0.162)=6.350, p<0.011, μ²=0.180). Further post-hoc paired t-tests with Bonferroni correction shows that 0-padding lead to statistically
better performance than Last-padding (p=0.012). This could be due to the fact that by padding with the last sample, the window labelled as non-protective would appear to contain guarding behavior (one of the protective behaviors) at the conclusion of the movement as the subject is maintaining the last position and ‘unable’ or ‘unwilling’ to move further. However, no significant differences were found with Next-padding (p=0.371, p=0.155). This is interesting as similar performances achieved by Next-padding suggest that a continuous modelling across the sequence of continuous movements could be feasible without requiring pre-segmentation, which can be a future direction of work.

6.2.3 Augmentation Methods. In Section 5.3, we present three data augmentation methods, namely reversing, gaussian noise and random discarding. For all the experiments conducted above, we combined the last two as such could lead to better performance than tested separately. Here we report the results achieved using different augmentations methods using Protect-LSTM (see Table 5). The hidden unit of each LSTM is set to 32, mini batch size is set respectively for each selected augmentation methods, the drop-out ratio at each layer is set to 0.5 while the weight is updated using the Adam algorithm [49].

|                          | Last-padding | Next-padding | 0-padding |
|--------------------------|--------------|--------------|-----------|
| LOSO                     | 0.7157       | 0.79         | 0.815     |
| Cross hold-out           | 0.693        | 0.685        | 0.728     |
| LSIO                     | 0.655        | 0.661        | 0.72      |
| p-value with Next-padding (<0.05) | .135        | -            | 0.371     |
| p-value with 0-padding (<0.05) | .012        | .371         | -         |

Table 4. Detection performance (F_{m}) under three padding methods with Protect-LSTM.

|                          | Without augmentation | Reversing | Gaussian noise (GN) | Random discarding (RD) | GN+RD |
|--------------------------|----------------------|-----------|---------------------|------------------------|-------|
| LOSO                     | 0.66                 | 0.395     | 0.686               | 0.66                   | 0.815 |
| Cross hold-out           | 0.55                 | 0.52      | 0.63                | 0.675                  | 0.728 |
| LSIO                     | 0.62                 | 0.528     | 0.668               | 0.678                  | 0.72  |
| p-value (<0.05)          | 0.003                | <0.001    | 0.006               | 0.001                  | -     |

Table 5. Detection performance (F_{m}) under three augmentation methods with Protect-LSTM.

We can see that the combination of the gaussian noise and the random discarding provide much better results than the other approaches and in the case of no augmentation. The reversing approach shows the worst performance in the case of our task. A repeated measures ANOVA showed significant difference in performance (LOSO mean F1 scores) between the augmentation methods: F(0,704,4)=6.697, p<0.001, \( \mu^2=0.39 \). The p-values computed for the post-hoc paired t-tests with Bonferroni correction between each method and the combination of gaussian noise and random discarding are also reported in Table 5. As a result, the combination of the Gaussian Noise and Random Discarding approaches lead indeed to statistically significantly better detection performances.

6.3 Impact of window length

The window segmentation approach is used with all the neural network algorithms. This is based on a window length of 3 seconds with overlapping ratio of 75%. In this subsection, we provide an analysis and justification for such choice. The boxplot in Fig 8 (a) show the duration of the movement instances in the EmoPain dataset.

As the figure shows there are strong differences between movement types and even between instances within the same movement, possibly due to different people’s physical and psychological capabilities. In particular, Reaching Forward shows big differences in duration among instances, possibly because the end point of the movement is much more affected by people’s capabilities than by the movement itself. [46] suggested that the window length need to be adjusted to different types of movement while the sliding step would be a tradeoff between the computation load and the segmentation accuracy. Consequently, an independent window length analysis is here conducted to investigate the performances of Protect-LSTM on different movement types.
Standing Reaching forward

Duration Distribution of Movement Instances

- 60 samples equal 1 second.

Fig. 8. (a) The duration distribution of movement instances in EmoPain database. As the sensors were operating at 60Hz, here 60 samples=1 second. (b) The Influence of Window length on protective behavior detection with different movement types. under different window lengths on protective behavior detection while using a fixed overlapping ratio of 75%.

To determine the best window length for the whole dataset which contains all the movements, another experiment on the whole dataset is presented in latter subsection.

6.3.1 Impact of Window Length within Movements. Similarly, as the cross hold-out validation described in Section 5.4, five experimental sets are made where each set only contains a single type of movement. Even though the duration of sit to stand and stand to sit is similar, they are still treated like separate instances and evaluated in two independent experimental sets. For each training set, the combined augmentation method (gaussian noise and random discarding) is applied.

The number of hidden units in each layer of the LSTM network is set to 32, the drop-out ratio is fixed to 0.5 and the mini batch size is adjusted according to the amount of data available for different movement types and different window lengths to minimize the overfitting tendency. The network weight is updated using the Adam algorithm [49]. Finally, the results of accuracy vs. window length is shown in Fig. 8 (b). We can conclude from the result that the window length has influenced the detection result (mean F1-score) to a noticeable degree for most movement types. The impact of window length on stand-on-one-leg is much smaller than others, which could be due to the characteristic of this type of movement where people kept standing still with one leg during most of the time, i.e., other than the initial raising and final descending of the movement, such movement simply requires to maintain the same position over a period of time. For the current experimental setting, the best window length for bending, standing on one leg, sit to stand, stand to sit and reaching forward is 5s/6s, 4s, 2.5s, 4s and 3s respectively (here 60 samples equal 1 second). Consequently, we choose to use the intermediate window lengths of 2.5s, 3s, 4s to explore the overall performance based on LOSO validation as to evaluate the impact of window lengths on different subjects across different movement types.

6.3.2 Impact of Window Length across Movement Types. Based on the experiment described above, we choose to investigate the impact of window lengths of 2.5s, 3s and 4s on the whole dataset with five movements pooled while an overlapping ratio of 75% is used. The validation method used here is LOSO validation, while the combined augmentation method (gaussian noise and random discarding) is also applied to each training sets. The hidden unit of each LSTM is set to 32, mini batch size is set respectively for each selected window lengths, the drop-out ratio at each layer is set to 0.5 while the weight is updated using the Adam algorithm [49]. The results are reported in Table 6 where the window length of 3s is found to produce the best overall results, which partially justified our usage of 3s sliding-window. A repeated measures ANOVA showed significant difference in performance (LOSO mean F1 scores) between the three window lengths: F(0.107, 1.322)=4.024, p=0.041, \( \mu^2=0.122 \). Further post-hoc paired t-tests with Bonferroni corrections on the mean F1-scores show that the 3s
window lead to statistically significantly better performances than the window of 4 seconds ($p=0.032$) and the difference approach significance comparing with the 2.5s window ($p=0.075$). No statistical differences existed between the performances achieved with the 4s and 2.5s windows.

In addition, specific results among the 30 subjects are reported in Fig. 9, where number 1 to 12 represent control subjects, 13 to 30 represent CLBP patients.

Table 6. Detection results (mean f1-score) on all movement types together using 3 different window lengths with Protect-LSTM and LOSO validation

| Window length | Bending | Standing on one leg | Sit to stand | Stand to sit | Reaching forward | All movements |
|---------------|---------|---------------------|--------------|--------------|-----------------|---------------|
| 2.5s          | 0.64    | 0.77                | 0.72         | 0.71         | 0.66            | 0.775         |
| 3s            | 0.75    | 0.8                 | 0.69         | 0.76         | 0.67            | 0.815         |
| 4s            | 0.75    | 0.81                | 0.66         | 0.83         | 0.67            | 0.731         |

![Impact of window length on individuals](image)

Fig. 9. Impact of window length on different subjects. 1-12: control subjects, 13-30 CLBP patients.

We can see from Fig. 9 that: i) the detection performances on most control subjects are 100% accurate which could be a result of the imbalanced distribution of training set where non-protective data represents a bigger proportion; ii) the detection results on patients fluctuate according to different window lengths, especially for subject 13, 16, 17, 22, 26, 28, 29 and 30, while the rest is less sensitive to window lengths. This suggest that individual differences are an important factor to consider in the modeling process.

6.3.3 Prediction on Single Timesteps. The training and testing conducted so far is all based on frames, while from [36] we learned that, for a continuous classification in HAR scenario, one could try to train the model with frames of variant lengths and do prediction on single timesteps. Here, to maintain the completeness of this work, we report the results (see Table 7) achieved with a similar approach, where frames generated from different sliding-windows (2.5s, 3s and 4s) are used for training and prediction is done on single timesteps. Protect-LSTM with all the three validation methods is used. The hidden unit of each LSTM is set to 32, mini batch size is fixed to 20, the drop-out ratio at each layer is set to 0.5 while the weight is updated using the Adam algorithm [49].

Table 7. Detection performance ($f_m$) under different training-testing strategies.

| Training and Testing Strategy                                    | LOSO | Cross Hold-out | LSIO | $P$-value (<0.05) |
|------------------------------------------------------------------|------|----------------|------|------------------|
| Frames of 3s length (default method)                             | 0.815| 0.735          | 0.74 |                  |
| Training with frames of 3s length while testing on single timesteps | 0.738| 0.62           | 0.61 | 0.039            |
| Training with frames of 2.5s, 3s and 4s lengths while testing on single timesteps | 0.836| 0.67           | 0.68 | 0.92             |

PACM Interact. Mob. Wearable Ubiquitous Technol, Vol. 0, No. 0, Article 0, Publication date: Month 2019.
From the results we can see that, i) training-testing on frames with optimal length lead to the best result for the detection of protective behavior in hold-out and LSIO validations; ii) training with windows of different lengths is better than using single window length when testing on single timestep and achieve the overall best result during LOSO validation, which implies the impact of frame lengths during training stage. A repeated measures ANOVA showed significant difference in performance (LOSO mean F1 scores) between the three methods: F(0.081, 2) = 8.645, p<0.002, \mu^2 = 0.23. Further post-hoc paired t-tests with Bonferroni corrections on the mean F1-scores (LOSO) show that training-testing on 3s frames is significantly better than training with 3s frames and testing on single timesteps (p=0.039), but no significance was found between the default method and training with three frame lengths and testing on single timesteps (p=0.92). Based on the unique characteristic of protective behavior recognition, the reason for such results can be the inadaptability of conducting prediction on a single timestep: i) protective behavior is exhibited in an intermittent way along with the execution of a specific movement, while the relevance of different types of protective behavior can be very small even within a same movement; ii) the labelling from experts was created by locating the onset sample and offset sample of a protective behavior (period) rather than deciding on each single timesteps, while the disagreement among labelers is enlarged by making the ground truth for a single timestep.

7 DISCUSSION

Our work in this paper was aimed at continuous (within-movement) detection of protective behavior from motion capture and sEMG data of people with CP and healthy controls from the EmoPain dataset [10], independently of the movement type. In our approach to addressing this problem, we used both convolutional and recurrent neural networks on 3-seconds data frames with strides of 0.75s (overlapping ratio of 75%) through instances of 5 different movement types performed by those participants. The best detection results were obtained with our Protect-LSTM: mean F1 score and accuracy of 0.82 and 0.87 respectively in LOSO validation. Further, we investigated the effect of window size on detection performance. Our findings show that the optimal window parameters vary between movement types as well as between individuals although 3s window with 75% overlap was found to be a common optimum.

Experiments with different padding approaches showed that 0-padding led to better results than Last-padding. It is possible that padding with the last sample in a frame wrongly conveys a state of movement freezing, which is a characteristic of guarding (one of the protective behaviors). However, possibly more interesting is that 0-padding and the Next-padding did not lead to statistically different results. This is important as our rehabilitation scenarios show that in most cases, no exercise sessions but every day functional activity is the focus of CP self-direct rehabilitation. Hence, we can expect that movements are not well defined, but they merge with each other as a person moves from one activity to another. Even if we have not addressed continuous modeling across movements and transitions, this result is encouraging to move to this next step. In terms of data augmentation methods, the combination of Gaussian Noise and Random Discarding approaches were found to lead to statistically significant better results than either of the two individual methods or the Reversing approach. In the rest of this section, we discuss these results in more depth highlighting the contributions that these findings make.

Our result of mean F1 score of 0.815 for fine-grained automatic detection of protective behavior during movement instances, lays the groundwork for the provision of real-time personalized support to people with CP during physical rehabilitation sessions. Unlike the state of the art in [10] where such behavior can only be detected after completion of movement instances and at instance level, our work enables detection of frames in which the protective behavior occurs within an instance of a movement. If combined with a movement detection system, our model can be used to deliver informed feedback during the execution of the movement or for further execution of such movement during situated exercise sessions or functional activities. For example, at maximal flexion during a forward reach, when a person with CP may guard by unhelpfully stiffening the lower back (as shown in Fig. 3 (a)) [22], our model can detect this behavior nearly as soon as it occurs, providing opportunity for just-in-time provision of encouragement to breathe deeply (so as to facilitate natural muscle relaxation) to the person, similar to what a clinician would do. Using another illustration, if the person...
demonstrate protective behavior at the start of a sit-to-stand, for instance, putting the feet forward and/or placing the hands on the seat for support (as shown in Fig. 3 (b)), our model can recognize this has occurred in about a few seconds after it does, enabling the technology to almost immediately suggest a more helpful strategy such as using a higher chair until confidence and affective capability is increased and allows for greater challenge. Studies on technology for CP physical rehabilitation (e.g. [8][6]) highlight the need to address the problems of engagement in functional movements. Findings in [13], which investigates the transfer of pain level detection capability from exercises to functional movements, suggest that our detection model can be further extended to functional settings where body movement data can be captured using low-cost wearable motion capture and sEMG sensors.

Our model performances are similar to findings in state-of-the-art [33] on continuous automatic detection of anomalous behavior based on motion capture data. Still, future work will consider the possibility of improving the performances of our model. One major challenge to be addressed is the crafting of learning architectures that address the complexity of human skeleton configuration (such as is found in motion capture data). Tailoring architectures to data types and format is an approach that has proved successful for image data [29][30][31][32], which occurs naturally in grid format that the convolutional layers are specifically designed to process [50]. One direction is to consider the prevalent motion capture format, biovision hierarchy [51], where joints are intuitively considered in hierarchical configuration starting from a root joint (usually the hip) to endpoints (the limbs). Nevertheless, our findings show that LSTM layers lead to much better than chance level detection for protective behavior on motion capture and sEMG data. The superiority of the LSTM over networks with convolutional layers seen in our findings is similar to findings in human activity recognition [33][36].

Our findings on the influence of sliding-window sizes on detection performance support similar findings in [46]; however, [46]’s investigation was based on much longer movement tasks, e.g. “walking in a supermarket while collecting items”, and they considered larger window sizes (from greater than 4s to about 1024s). In this paper, we look at movement on a finer scale, with movements, such as in sit-to-stand, as brief as 1.05s on average. More importantly, the classification task in our work is of transient affective behavior, unlike the goal of [46] on activity recognition. We found that the detection performance increased with the size of the sliding-window used until a certain peak beyond which performance dropped. This could be because shorter frame length may fail to provide sufficient temporal-continuous information, while the decrease in performance with larger sliding-window sizes may be due to the consequent smaller number of frames for training produced by such segmentation. On the other hand, as proposed in [36] and with larger dataset under HAR scenario, using dynamic window lengths for bootstrapping and do prediction on a single timestep can also be a direction instead of selecting window parameters manually. But according to our current result and the existing labelling nature of protective behavior, it is better to use frame with considerable length as basic unit for modelling.

One interesting finding was, during the discussion of sliding-window sizes, the observation of many-peaks for movements like sit-to-stand and bending in the EmoPain dataset [10], although with one peak superior to the others. The many-peaks may indicate that the movement has two (non-consecutive) periods that provide critical information about protective behavior, with behaviors like stiffness and bracing both in starting phase and ending phase. This can be informative for improving continuous detection of protective behavior, as the position of a given window (from initiation of the movement) can be fed as an additional feature into the machine learning model. In fact, this may be similar to the strategy used by physiotherapists in judging pain behavior in consultation settings: based on a mental ‘list’ of periods during a movement where they watch for anomalous behavior.

Finally, in this work, we have taken a major-voting approach to define the ground truth and address the sparse occurrence of protective behavior samples within a frame. For real-life deployment of such technology, a more relax or more conservative approach could be used to label a frame. This could be based, for example, on the person’s overall capability to self-manage his/her condition and engage in physical or functional activity. In [6], physiotherapists state that they tend to intervene more frequently during early pain management sessions and gradually let the person to take responsibility to overcome barriers and intervene only if the person start to become more anxious (e.g. longer or closely repeated protective behavior). In addition, other learning techniques that are specifically designed to deal with label sparsity within frames (e.g., Multiple Instance
Learning [63] [64]) could be explored and possibly integrated in our approach to remove the need for building a frame-level ground truth but still taking into account that, even if the labels are assigned at each single timestep, the rater’s judgment is based also on the dynamics emerging between consecutive timesteps.

8 CONCLUSION
In this paper, we investigated the possibility of automatically detecting frames of protective behavior in people suffering from CLBP during psychologically and physically demanding movement. The aim is to develop movement- and muscle activity-recognition 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 (motion capture and EMG data), we show that the Protect-LSTM network performs much better than the CNN- and RF-based models: mean F1 score of 0.815 (LOSO validation), leading to excellent agreement with average expert raters. Indeed, the results also show that Protect-LSTM performs better than bi-LSTM and the Dual-stream LSTM model even if the differences were not statistically significant. Another proposed model called Dual-stream LSTM performed similar to Protect-LSTM in LOSO and LSIO validations while achieved better results in cross hold-out validation, which showed the good potential of using better architecture that pay respect to different data modalities and structure.

Rather than using directly the raw data, we made use of low-level features. These features together with our data augmentation methods overcome the problem of a small dataset when using deep learning as they carry more information than raw data.

We additionally provide evidence of the impact of sliding-window length on detection performance and we suggest that these choices 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. For the 5 movements available in the EmoPain database [10], we showed that windows of length 3 and 75% overlap provide best performances. Better performances could be reached for individual movement models to be built (e.g., during situated predefined movement) and by considering instances from the specific individual. More research will be directed to combine the detection of protective movement with the detection of the type of movements and to move towards full continuous detection of protective behavior across functional activity rather than instances of movement only.

ACKNOWLEDGMENTS

REFERENCES

[1] Breivik, H., Collett, B., Ventafridda, V., Cohen, R., & Gallacher, D. Survey of chronic pain in Europe: Prevalence, impact on daily life, and treatment. European Journal of Pain, 10(4), 287–333, (2006).
[2] Tracey, L., & Bushnell, M. C. How Neuroimaging Studies Have Challenged Us to Rethink: Is Chronic Pain a Disease? Journal of Pain, 10(11), 1113–1120, (2009).
[3] Keefe, F. J., & Block, A. R. Development of an observation method for assessing pain behavior in chronic low back pain patients. Behavior Therapy, 13(4), 363–375, (1982).
[4] Vlaeyen, J. W. S., & Linton, S. J. Fear-avoidance and its consequences in chronic musculoskeletal pain: A state of the art. Pain, 85(3), 317–332, (2000).
[5] Vlaeyen, J. W. S., Morley, S., & Crombez, G. The experimental analysis of the interruptive, interfering, and identity-distorting effects of chronic pain. Behaviour Research and Therapy, 86, 23–34, (2016).
[6] Singh, A., Klapper, A., Jia, J., Fidalgo, A., Jimenez, A. T., Kanakam, N., Williams, A. Motivating People with Chronic Pain to do Physical Activity: Opportunities for Technology Design. In International Conference on Human Factors in Computing Systems (CHI) (pp. 2803–2812, (2014).
[7] Singh, A., Piana, S., Pollarolo, D., Volpe, G., Varni, G., Tajadura-Jimenez, A., Bianchi-Berthouze, N. 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), 1–49, (2016).
[8] Singh, A., Bianchi-Berthouze, N., & Williams, A. C. Supporting Everyday Function in Chronic Pain Using Wearable Technology. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, 3903–3915, (2017).
[9] Fordyce, W. E., Lansky, D., Calsyn, D. A., Shelton, J. L., Stolov, W. C., & Rock, D. L. Pain measurement and pain behavior. Pain, 18(1), 33–69, (1984).
[10] Aung, MSH, et al. "The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal EmoPain dataset." IEEE transactions on affective computing 7.4 (2016): 435-451.
[11] Olugbade, T. A., Aung, M. S. H., Bianchi-Berthouze, N., Marquardt, N., & Williams, A. C. Bi-Modal Detection of Painful Reaching for Chronic Pain Rehabilitation Systems. Proceedings of the 16th International Conference on Multimodal Interaction - ICM '14, 455–458, (2014).
[12] Olugbade, T. A., Bianchi-Berthouze, N., Marquardt, N., & Williams, A. Pain Level Recognition using Kinematics and Muscle Activity for Physical Rehabilitation in Chronic Pain. Proc. 6th Conf, 243–249, (2015).
[13] Olugbade, T. A., Bianchi-Berthouze, N., Marquardt, N., & Williams, A. C. Human Observer and Automatic Assessment of Movement Related Self-Efficacy in Chronic Pain: from Movement to Functional Activity. IEEE Transaction on Affective Computing, (2018).
[14] Sullivan, M. J. L., Thibault, P., Savard, A., Catchlove, R., Kozev, J., & Stanish, W. D. The influence of communication goals and physical demands on different dimensions of pain behavior. Pain, 125(3), 270–277, (2006).
[15] de Gelder, B. Why bodies? Twelve reasons for including bodily expressions in affective neuroscience. Philosophical Transactions of the Royal Society B: Biological Sciences, 364(1535), 3475–3484, (2009).
[16] Kleinsmith, Andrea, and Nadia Bianchi-Berthouze. "Affective body expression perception and recognition: A survey." IEEE Transactions on Affective Computing 4.1 (2013): 15-33.
[17] Kleinsmith, Andrea, Nadia Bianchi-Berthouze, and Anthony Steed. "Automatic recognition of non-acted affective postures." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 41(4) (2011): 1027-1038.
[18] The Pain Consortium. (2016). UK Pain Messages. Pain News, 21–22.
[19] Jamison, R. N. (2016). Are we really ready for telehealth cognitive behavioral therapy for pain? Pain, 0(0), 1.
[20] Cook, K. F., Keefe, F., Jensen, M. P., Roddey, T. S., Callahan, L. F., Revicki, D., ... Attmann, D. (2013). Development and validation of a new self-report measure of pain behaviors. Pain, 154(12), 2867–2876.
[21] Watson, F. J., Booker, C. K., & Main, C. J. (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.
[22] Ahern, D. K., Follick, M. J., Council, J. R., Laser-Wolston, N., & Litchman, H. (1988). Comparison of lumbar paravertebral EMG patterns in chronic low back pain patients and non-patient controls. Pain, 34(2), 153–160.
[23] Grip, H., Ohberg, F., Wöldt, U., Sterner, Y., Karlsson, J. S., & Gerdle, B. (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.
[24] Dickey, J. P., Pierrynowski, M. R., Bednar, D. a., & Yang, S. X. (2002). Relationship between pain and vertebral motion in chronic low-back pain subjects. Clinical Biomechanics, 17(5), 345–352.
[25] Aung, MSH, et al. "Automatic recognition of fear-avoidance behavior in chronic pain physical rehabilitation." Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), (2014).
[26] Miotto, Riccardo, et al. "Deep learning for healthcare: review, opportunities and challenges." Briefings in bioinformatics (2017).
[27] Lucey, Patrick, et al. "Painful data: The UNBC-McMaster shoulder pain expression archive database." Automatic Face & Gesture Recognition and Workshops (FG 2011). 2011 IEEE International Conference on. IEEE, (2011).
[28] Prkachin, Kenneth M., and Patricia E. Solomon. "The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain." Pain 139.2 (2008): 267–274.
[29] Bellantonio, Marco, Mohammad A. Haque, Pau Rodriguez, Kamal Nasrollahi, Taisi Telve, Sergio Escalera, Jordi Gonzalez, Thomas B. Møeslund, Fejmian Rasti, and Gholamreza Anbarjafari. "Spatio-temporal pain recognition in cun-based super-resolved facial images." In Video Analytics. Face and Facial Expression Recognition and Audience Measurement, pp. 151-162. Springer, Cham, 2016.
[30] Zhou, Yuqian, Jinmin Pi, and Bertram E. Shi. "Pose-independent facial action unit intensity regression based on multi-task deep transfer learning." In Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on, pp. 872-877. IEEE, 2017.
[31] Wang, Feng, Xiang Xiang, Chang Liu, Trac D. Tran, Austin Reiter, Gregory D. Hager, Harry Quon, Jian Cheng, and Alan L. Yuille. "Regularizing Face Verification Nets For Pain Intensity Regression." arXiv preprint arXiv:1702.06925 (2017).
[32] Rodriguez, Pau, Guillem Cucurull, Jordi Gonzalez, Josep M. Gonfaus, Kamal Nasrollahi, Thomas B. Møeslund, and F. Xavier Roca. "Deep pain: Exploiting long short-term memory networks for facial expression classification." IEEE transactions on cybernetics (2017).
[33] Hammerla, Nils Y., Shane Halloran, and Thomas Ploetz. "Deep, convolutional, and recurrent models for human activity recognition using wearables." arXiv preprint arXiv:1604.08880 (2016).
[34] Chavarriaga, Ricardo, et al. "The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition." Pattern Recognition Letters 34.15 (2013): 2033-2042.
[35] Reiss, Attila, and Didier Stricker. "Introducing a new benchmarked dataset for activity monitoring." Wearable Computers (ISWC), 2012 16th International Symposium on. IEEE, (2012).
[36] Guan, Yu, and Thomas Ploetz. "Ensembles of deep lstm learners for activity recognition using wearables." Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1.2 (2017): 11.
[37] Daniel Roggen, Gerhard Tröster, Georg Ogris, and Paul Lukowicz. "Wearable activity tracking in car manufacturing." IEEE Pervasive Computing 7, 2 (March 2008), 42–50.
Automatic Detection of Protective Behavior in Chronic Pain Physical Rehabilitation: A Recurrent Neural Network Approach

Morales, Francisco Javier Ordóñez, and Daniel Roggen. "Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations." Proceedings of the 2016 ACM International Symposium on Wearable Computers. ACM, 2016.

Bachlin, Marc, et al. "Potentials of enhanced context awareness in wearable assistants for Parkinson's disease patients with the freezing of gait syndrome." Wearable Computers, 2009. ISWC’09. International Symposium on. IEEE, 2009.

Karg, M., Samadani, A. A., Gorbet, R., Kuhnlenz, K., Hoey, J., & Kulic, D. (2013). Body movements for affective expression: A survey of automatic recognition and generation. IEEE Transactions on Affective Computing, 4(4), 341–359.

Rad, Nastaran Mohammadian, and Cesare Furlanello. "Applying deep learning to stereotypical motor movement detection in autism spectrum disorders." Data Mining Workshops (ICDMW), IEEE 16th International Conference on. IEEE, (2016).

Albinali, Fahd, Matthew S. Goodwin, and Stephen S. Intille. "Recognizing stereotypical motor movements in the laboratory and classroom: a case study with children on the autism spectrum." Proceedings of the 11th international conference on Ubiquitous computing. ACM, (2009).

Goodwin, Matthew S., et al. "Moving towards a real-time system for automatically recognizing stereotypical motor movements in individuals on the autism spectrum using wireless accelerometry." Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, (2014).

Cicchetti D V. Guidelines, Criteria, and Rules of Thumb for Evaluating Normed and Standardized Assessment Instruments in Psychology. Psychological Assess. 1994;6(4):284–90.

McGraw, K. O., & Wong, S. P. Forming inferences about some intraclass correlation coefficients. Psychological methods, 1996, 1(1), 30.

Huynh, Tâm, Ulf Blanke, and Bernt Schiele. "Scalable recognition of daily activities with wearable sensors." In LoCa, vol. 7, pp. 50-67. 2007.

Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

Greff, Klaus, et al. "LSTM: A search space odyssey." IEEE transactions on neural networks and learning systems 28.10 (2017): 2222-2232.

Kim, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." 2014 ICLR.

LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

Meredith, Maddock, and S. Maddock. "Motion capture file formats explained." Department of Computer Science, University of Sheffield 211 (2001): 241-244.

Papi, Enrica, Athina Belsi, and Alison H McGregor. "A Knee Monitoring Device and the Preferences of Patients Living with Osteoarthritis: A Qualitative Study." BMJ Open, 5 (9). (2015).

Papi E, Murtagh GM, McGregor AH. "Wearable technologies in osteoarthritis: a qualitative study of clinicians' preferences". BMJ OPEN, Vol. 6, (2016): 2044-6055.

Barbareschi G, Holloway CSM, Berthouze N, Sonenblum S, Sprigle S. "Use of a Low Cost, Chest-Mounted Accelerometer to Evaluate Transfer Skills of Wheelchair Users During Everyday Activities". JMIR Rehabilitation and Assistive Technologies, Vol.26, (2018).

Turk, Dennis C., and Akiko Okifuji. "Psychological factors in chronic pain: Evolution and revolution." Journal of consulting and clinical psychology, Vol .70 (3), (2002).

Bagalkot Naveen L., Tomas Sokoler, and Suraj Badkar. "Reride: performing lower back rehabilitation while riding your motobike in traffic." Proceedings of the 10th EAI international Conference on Pervasive Computing Technologies for Healthcare. (2016).

Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. Comput. Surveys 46, 3 (Jan. 2014), 1–33.

Hallgren KA. Computing inter-rater reliability for observational data: an overview and tutorial. Tutorials in quantitative methods for psychology, 2012,8(1):23.

Wang, Limin, Yu Qiao, and Xiaoou Tang. "Action recognition with trajectory-pooled deep-convolutional descriptors." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4305–4314. 2015.

Ha, Sojeong, and Seungjin Choi. "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors." In Neural Networks (IJCNN), 2016 International Joint Conference on, pp. 381-388. IEEE, 2016.

Yun W, Lee D, Park C, Kim J, Kim J. Automatic Recognition of Children Engagement from Facial Video using Convolutional Neural Networks. IEEE Transactions on Affective Computing, 2018, 3045:1–12.

Luca Romeo, Andrea Cavallo, Nadia Bianchi-Berthouze, Massimiliano Pontil. Multiple Instance Learning for Emotion Recognition using Physiological Signals. IEEE Transaction in Affective Computing, in review.

K. Sikka, A. Dhall, and M. S. Bartlett, "Classification and weakly supervised pain localization using multiple segment representation," Image and Vision Computing, 32(10), 659 – 670, 2014.

A. Ruiz, O. Rudovic, X. Binefa, and M. Pantic, "Multi-Instance Dynamic Ordinal Random Fields for Weakly-supervised Facial Behavior Analysis," ArXiv e-prints, Feb. 2018.