Combining Real-Time Segmentation and Classification of Rehabilitation Exercises with LSTM Networks and Pointwise Boosting

Antonio Bevilacqua,[1] **Giovanni Ciampi,**[2] **Rob Argent,**[1,3] **Brian Caulfield,**[1] **Tahar Kechadi**[1]
[1]Insight Centre for Data Analytics, University College Dublin, [2]Università degli studi di Salerno, [3]Beacon Hospital, Dublin
{antonio.bevilacqua, rob.argent, brian.caulfield, tahar.kechadi}@insight-centre.org, g.ciampi5@studenti.unisa.it

Abstract

Autonomous biofeedback tools in support of rehabilitation patients are commonly built as multi-tier pipelines, where a segmentation algorithm is first responsible for isolating motion primitives, and then classification can be performed on each primitive. In this paper, we present a novel segmentation technique that integrates on-the-fly qualitative classification of physical movements in the process. We adopt Long Short-Term Memory (LSTM) networks to model the temporal patterns of a streaming multivariate time series, obtained by sampling acceleration and angular velocity of the limb in motion, and then we aggregate the pointwise predictions of each isolated movement using different boosting methods. We tested our technique against a dataset composed of four common lower-limb rehabilitation exercises, collected from heterogeneous populations (clinical and healthy). Experimental results are promising and show that combining segmentation and classification of orthopaedic movements is a valid method with many potential real-world applications.

1 Introduction

Integrating machine learning methods in healthcare applications is becoming an established trend within the context of rehabilitative science (Tack 2019). Patients and practitioners can both benefit from the adoption of autonomous biofeedback tools, capable of producing quantitative and qualitative feedback during physical therapy, with the aim of supporting patient adherence (Argent, Daly, and Caulfield 2018; Giggins, Persson, and Caulfield 2013). A common pipeline to generate such feedback involves a segmentation step, that is, the isolation of individual movements from a longer activity stream, and a subsequent classification step, so that deviant or aberrant movements can be identified and notified to the patients. Often, systems that acquire the input data from inertial sensors such as accelerometer and gyroscope (Godfrey et al. 2008) address segmentation and classification as completely distinct, disjoint steps. In this paper, we present a novel segmentation technique that embeds on-the-fly classification for rehabilitation movements, sampled as multivariate timeseries of inertial signals, within the segmentation process. Our method works on streaming data, thus it provides live feedback to the users, and does not require domain knowledge about the target exercises. These two aspects constitute crutial requirements for any biofeedback system to be adopted in real world applications.

The rest of the paper is organised as follows. Section 2 briefly introduces existing segmentation and classification methods currently in the state of the art. In Section 3 we lay out a formal description of the segmentation task, and we provide an overview of our proposed solution. In Section 4 we introduce our datasets and the basic processes behind data collection and annotation. Findings of the experimental campaigns are presented in Section 5, while final remarks and considerations are given in Section 6.

2 Related Work

An extensive body of literature exists about human motion segmentation, and specifically rehabilitation exercise segmentation and classification. In their comprehensive study, Lin, Karg, and Kulić introduce the problem of segmentation in the context of human motion modeling (Lin, Karg, and Kulić 2016). The authors describe the main requirements of a segmentation system, and introduce common state-of-the-art segmentation techniques such as Zero Velocity Crossing (ZVC) or Hidden Markov Models (HMM). ZVC is based on the detection of points where there is a change of direction in the movement (Fod, Matarić, and Jenkins 2002), while HMMs model the observed signals as a sequence of unobservable Markovian states. Both ZVC and HMM are popular in literature, as they do not rely on domain knowledge. However, given their poor mechanism of false positive rejection, they are prone to oversegmentation. In a study conducted by Bevilacqua et al., the authors propose a segmentation technique based on the clustering of ZVC points. The cluster centroids are used to extract segment candidates, and analytically adjusted until a good match is found against previously generated templates. Once segments are isolated, they are classified using a set of features extracted from the raw input signals (Bevilacqua et al. 2018). This method was proved to be effective on a cohort of exercises collected in a laboratory environment, however, it requires the entire exercise to be completed before segmentation and classification can take place. More recently, Lin, Joukov, and Kulić suggested a segmentation strategy based on the classification of individual points of the joint angle measure (Cheng and Oel-

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

mann 2010) as either segment points or non-segment points (Lin, Joukov, and Kulić 2018). The authors apply PCA to the input signals in order to reduce the degrees of freedom (DoF) of the data and hence the computational demand of the technique. As a tradeoff, also this technique is unsuitable for working on a live stream of data. To overcome this issue, Bevilacqua et al. designed convFSM, a segmentation approach that involves the classification of sliding windows extracted from the signal using a Convolutional Neural Network (CNN). The predictions obtained from the CNN are then fed into a Finite State Machine (FSM) that models the temporal state of the signal and, therefore, the exercise execution (Bevilacqua et al. 2019). This technique minimizes the required domain knowledge, but heavily relies on the assumption that the streaming signal starts from a condition of resting, which is not always true in unsupervised, real-world scenarios. Moreover, given the nature of the FSM, it is not suitable for modeling isokinetic motion patterns.

The purpose of our novel segmentation technique is then to overcome the main gaps left open by existing approaches, with respect to domain knowledge, false positive rejection, and capability of working live on data streams.

3 Proposed Approach

Our feedback system is based on the main assumption that segmentation and classification of exercise primitives are intertwined steps, and should not be addressed in isolation. In order to enforce such assumption, we designed a LSTM network responsible of producing label sequences for the input signals. We can then aggregate these labels in order to extract both the edge points for the motion primitives and the final qualitative feedback for each one of them. A high-level overview of the system is depicted in Figure 1, where we illustrate the segmentation/classification pipeline starting from the signals being streamed by the sensors.

Problem formulation

An exercise is a streaming sequence S of multidimensional points $\mathbf{s_i}$ sampled from one or more inertial sensors. The dimensionality of these points depends on the sensors being used for the sampling (for instance, points sampled from a triaxial accelerometer will have coordinates $[a_x, a_y, a_z]$). The stream subsequence between time $t-k$ and time t is denoted with S_{t-k}^{t}. As depicted in Figure 1, our LSTM network accepts in input subsequences of type S_{t-k}^{t}, where the number k of timesteps in the sequence and the stride value between consecutive windows, are hyperparameters. For each input sequence, the network produces a probability vector $\mathbf{p_t} = [p_t^s, p_t^e, p_t^c]$, where p_t^s is the probability of the signal to be silent at time t (no physical motion is detected at that particular time), p_t^e is the probability of the signal to correspond to physical movement at time t when the detected movement is classified as deviant or far from a correct execution, and p_t^c is the probability of the signal to correspond to physical movement at time t when the detected movement is classified as correctly executed. Input sequences for $t < k$ are padded with zeros, so every point $\mathbf{s_t}$ of the stream is assigned to a probability vector $\mathbf{p_t}$. During an exercise execution, the patient performs a number of physical repetitions, also called *motion primitives*. The main objective of the segmentation algorithm is to isolate all the motion primitives in the time domain, while the exercise is being executed. An example is illustrated in Figure 2, where 4 motion primitives are performed. The vertical dashed lines in the picture correspond to edge points marking the beginning and the end of each primitive, while the solid coloured bars on top of the graph indicate the class assigned to each point of the stream.

Segment detection

Transitions in the streaming signal correspond to physical changes of state of the limb in motion. As rehabilitation exericses comprise multiple executions of the same movement, we are interested in isolating all the points in time where a movement starts or ends. In order to do so, we apply Algorithm 1 to the sequence S.

Algorithm 1: Segmentation algorithm

```
Input  : Streaming signal $S$
Input  : Number $k$ of LSTM input timesteps
Input  : Number $n$ of previous probability values
Output: Set of cutting points for the input signal
1  coordinates $\leftarrow$ [ ]
2  lstm_win $\leftarrow [\mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}]$ FIFO queue, fixed length $k$
3  sma_win $\leftarrow$ [ ] FIFO queue, maximum length $n$
4  rest $\leftarrow True$
5  while new point $\mathbf{p_t}$ available from S do
6      push $\mathbf{p_t}$ in lstm_win
7      $[p_t^s, p_t^e, p_t^c]$ = LSTM(lstm_win)
8      push $p_t^s$ in sma_win
9      $m$ = MEAN(sma_win)
10     if rest and $((m < p_t^e)$ or $(m < p_t^c))$ then
11         push $t$ in coordinates
12         rest $\leftarrow False$
13     else if not rest and $(m > p_t^e)$ and $(m > p_t^c)$ then
14         push $t$ in coordinates
15         rest $\leftarrow True$
16 subtract $n/2$ from all items in coordinates
17 return coordinates
```

In short, through our segmentation algorithm, a probability vector $\mathbf{p_t}$ is produced for every point of the stream S (line 7), and the values of p_t^s are stored in a FIFO queue of maximum length n. After that, the average of the last n probability values p_i^s is computed (line 9). If the signal is currently in a state of silence, and the computed average is smaller than p_t^e or p_t^o, then a starting coordinate is extracted, and the signal is considered to be in a state of motion (lines 10-12). However, if the signal is currently in a state of motion, and the computed average is bigger than both p_t^e and p_t^c, then an ending coordinate is extracted at time t, and the signal is marked to be in a state of silence (lines 13-15). The windowing mechanism used for the average computation of the probability values p_i^s is an implementation of the

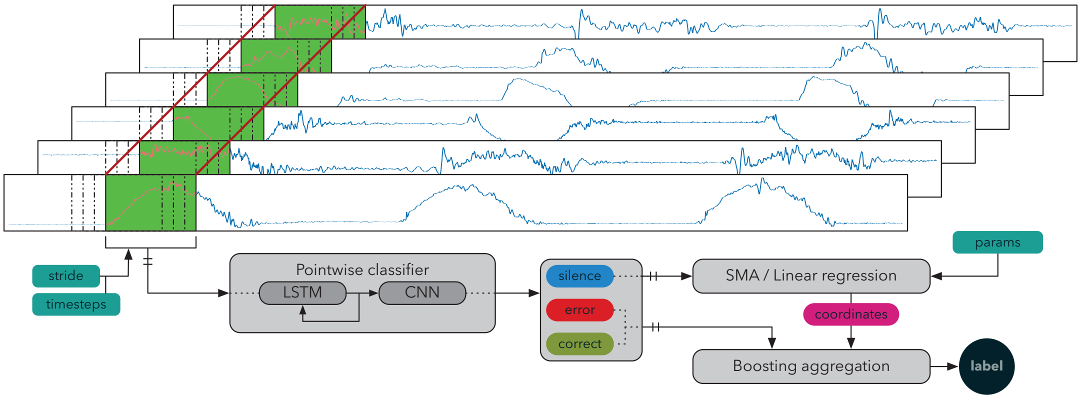


Figure 1: High-level overview of the system. The input stride and number of timesteps, and the smoothing parameters for the probability sequences are provided to the system after empirical testing.

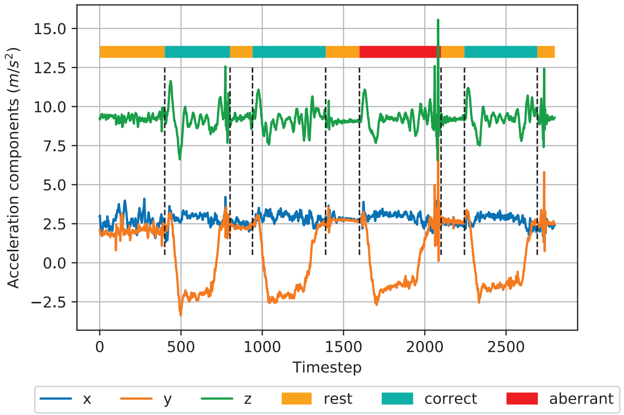


Figure 2: Signals sampled from an exercise execution.

Simple Moving Average (SMA). Applying the SMA rather than using the raw probability values has a smoothing effect on the probability subsequence, reducing oversegmentation in regions of confusion often occurring when a movement starts and when a movement ends. This is shown in Figure 3a, where the raw sequence of values $\mathbf{p^s}$ is drawed alongside its smoothed version $SMA(\mathbf{p^s})$.

Segment classification: prediction boosting

A motion primitive starting at time a and ending at time b corresponds to the input sequence S_a^b, and is mapped by the LSTM network to a sequence of probability vectors P_a^b, as described by Algorithm 1. We use for P the same notation introduced for the sequence S, in order to analytically aggregate its values and produce a final qualitative label for the motion primitive. For the sequence P_a^b, however, we do not take into account the temporal ordering of its values, and we only consider the values of $\mathbf{p^e}$ and $\mathbf{p^o}$, as illustrated in Figure 3b. We apply to the isolated primitives four different aggregation techniques, three of which were originally designed by Toto, Foley, and Rundensteiner for audio clip classification (Toto, Foley, and Rundensteiner 2019).

Deviation Voting (DV) The class $\hat{d}$ assigned to the target primitive is the most frequent predicted class among all the points in the primitive. This is formally expressed in Equation 1, where the compressed notation $p_i = \bar{p}$ indicates that for point i in the sequence, the highest probability of the network softmax output has the same index of $\bar{p}$.

$$\hat{d} = \underset{\bar{p}\in\{p_e,p_o\}}{\operatorname{argmax}} \left(|\{p_i : p_i = \bar{p}, \forall p_i \in P_a^b\}|\right) \qquad (1)$$

This aggregation method is based on the assumption that the qualitative characterisation of a movement is evenly distributed across its full duration, so all timesteps equally contribute to the final prediction.

Cumulative Deviation Strength (CDS) The class $\hat{d}$ of a repetition is selected as the class with the highest cumulative probability across all its point, as shown in Equation 2.

$$\hat{d} = \underset{\bar{p}\in\{p_e,p_o\}}{\operatorname{argmax}} \left(\sum_{p_i \in P_a^b : p_i = \bar{p}} p_i \right) \qquad (2)$$

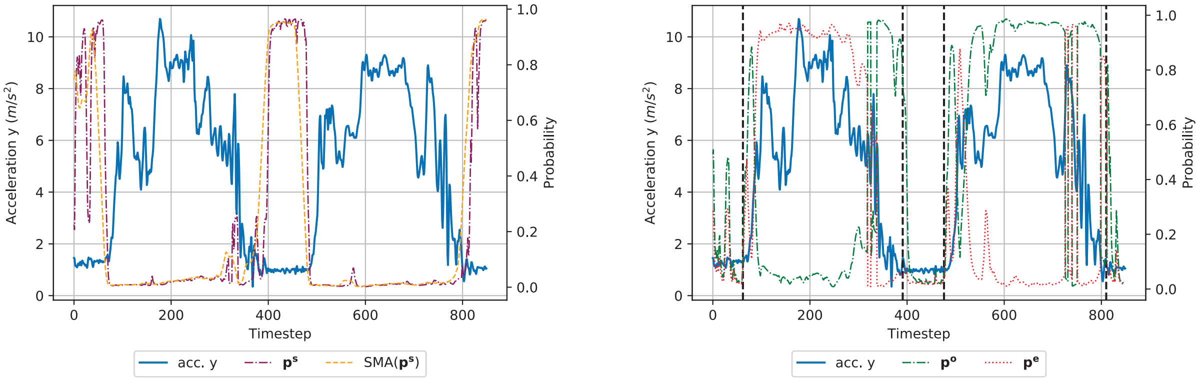
The rationale behind CDS is the same as DV, but in this case also the degree of confidence of the pointwise labels is taken into account. CDS can be interpreted as weighted DS.

Maximum Deviation Strength (MDS) The class $\hat{d}$ of a primitive is selected as the class with the highest probability among all the probability values in P_a^b, as per Equation 3.

$$\hat{d} = \underset{\bar{p}\in\{p_e,p_o\}}{\operatorname{argmax}} \left(\max\left(p_i : p_i = \bar{p}, \forall p_i \in P_a^b\right)\right) \qquad (3)$$

MDS works under the assumption that a full movement characterisation can be extracted from an individual point in the sequence, for which the network returned the highest overall probability.

***q-best* Deviation Rank (qDR)** The class $\hat{d}$ of a primitive is selected as the most frequent class among the q classes with the highest probability values in P_a^b. When $q = 1$, qDR corresponds to MDS, while $q = |P_a^b|$ makes qDR equivalent to DV. In our experiments, we set q to 30 (30DR), 50 (50DR), and half the length of the target primitive (halfDR).



(a) Signal segmentation

(b) Segment classification

Figure 3: Segmentation and classification with the probability values generated by the LSTM network.

Table 1: Dataset composition

	exercise	subjects	sets	reps	correct (%)
train	HS	49	94	1055	0.65
	SKE	47	94	1043	0.43
	IRQ	45	84	924	0.54
	SLR	49	89	986	0.53
test healthy	HS	10	10	148	1
	SKE	10	10	150	1
	IRQ	10	10	150	1
	SLR	10	10	150	1
test clinical	HS	10	23	320	1
	SKE	11	17	241	0.43
	IRQ	10	18	270	0.75
	SLR	10	21	297	0.49

4 Data and Methodology

In this work we target 4 lower limb rehabilitation exercises, namely, heel slide (HS), seated knee extension (SKE), inner range quadriceps (IRQ), and straight leg raise (SLR), which are commonly undertaken following orthopaedic surgery (American Academy of Orthopaedic Surgeons 2017). For each exercise, we collected a training set and 2 separate, independent test sets. Table 1 lists the number of participants, sets, and repetitions for each exercise and dataset. The percentage of correct repetitions is included. All the exercises are collected with the Shimmer inertial unit (Burns et al. 2010), a research device that embeds a number of inertial sensors. For this work, we use the accelerometer, with a range of $\pm 2g$, and the gyroscope, with a range of $500dps$. Both sensors sample data at a frequency of $102.4Hz$.

Training data

Training data were collected from a heterogeneous group of roughly 50 subjects, both in a controlled environment from healthy participants, and in a clinical setup from real-world patients. The proportion of clinical subjects to healthy subjects is approximately 1 to 4. A Chartered Physiotherapist was responsible of supervising the exercise executions in order to provide reliable ground truth for both the quantitative and qualitative characterisation of the data. Subjects can perform the exercises in a correct fashion, or with particular deviations caused by pain, distress, or fatigue (Bevilacqua et al. 2019). Whilst healthy subjects are instructed by the supervising Physiotherapist to artificially emulate a deviation, real-world patients perform the exercises to the best of their capabilities, thus providing genuine, aberrant examples, with the inclusion of other common physical imperfections such as co-articulations or baseline drifting.

Test data

As we use the training data for hyperparameter tuning, a separate test dataset was collected and used for performance evaluation. The test dataset is composed of exercises sampled from 10 healthy subjects, and 10 clinical subjects, as outilned in Table 1. All exercises in the test dataset are collected in a real-world environment, under the supervision of a Chartered Physiotherapist. Participants were not instructed to perform the exercises inducing specific deviations, so the resulting dataset for HS does not include any example of the negative class. Given the different quality of physical performance between healthy and clinical subjects, the results of these two groups are presented separately.

5 Experimental Results

For our experimental campaigns, we perform hyperparameter tuning on the training dataset, focusing on the number of timesteps and the stride between consecutive input windows. Tested values for the timesteps are 10, 20, 50, 100, and 250. Tested values for the stride are 10, 5, and 1. We adopt the Leave-One-Subject-Out (LOSO) validation protocol (Jordao et al. 2018), in order to avoid the introduction of positive bias while searching for the best hyperparameter configuration. With LOSO, validation splits are constructed so that data collected from the same subject cannot appear in both training and validation set at the same time. Figure 4 shows the breakdown of accuracy (4a) and loss (4b) for all the hyperparameter configurations. What stands out from this set of results is the increase in pointwise classification accuracy for long timestep inputs to the LSTM. The stride

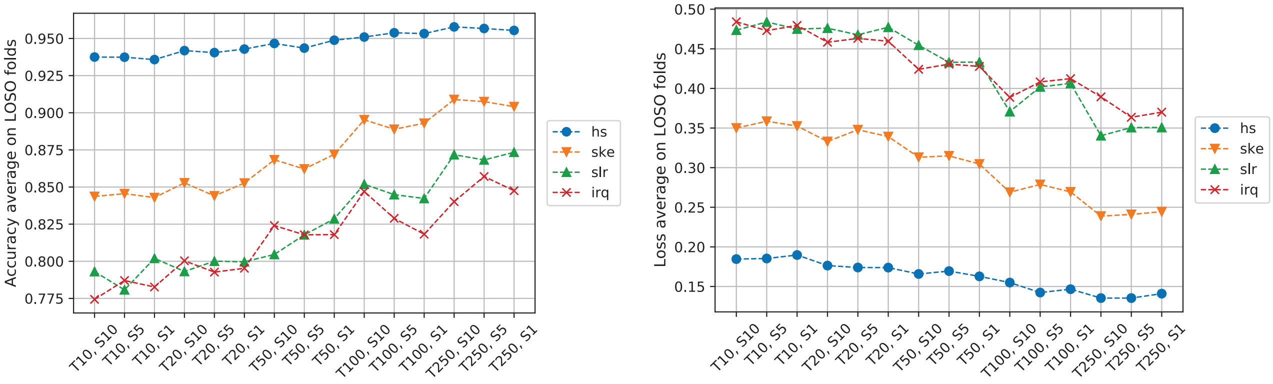
Table 2: Segmentation scores on test data, $t_{err} = \pm 35$, $n = 25$

(a) Healthy set

model	metric	LSTM	convFSM	clustree	hmm	zvc
	accuracy	0.9732	0.7507	0.7878	0.706	0.6892
HS	precision	0.9897	0.8605	0.8845	0.8277	0.7521
	recall	0.9831	0.8547	0.8783	0.8277	0.8918
	accuracy	1	0.9933	0.5463	0.8018	0.825
SKE	precision	1	1	0.7066	0.89	0.8319
	recall	1	0.9933	0.5605	0.89	0.99
	accuracy	0.9672	0.7758	0.85	0.5582	0.6082
SLR	precision	0.9833	0.8708	0.9315	0.7301	0.7283
	recall	0.9833	0.8766	0.9066	0.7033	0.7866
	accuracy	0.9638	0.8098	0.9354	0.2625	0.5179
IRQ	precision	0.9865	0.9103	0.9666	0.8076	0.8357
	recall	0.9766	0.88	0.9666	0.28	0.5766

(b) Clinical set

model	metric	LSTM	convFSM	clustree	hmm	zvc
	accuracy	0.8466	0.2828	0.5362	0.2815	0.5505
HS	precision	0.8654	0.7674	0.7737	0.4397	0.5932
	recall	0.975	0.3093	0.6359	0.439	0.8843
	accuracy	0.9151	0.8772	0.5379	0.7629	0.6002
SKE	precision	0.9721	0.9506	0.7418	0.9068	0.6528
	recall	0.9398	0.919	0.6618	0.8278	0.8817
	accuracy	0.7138	0.6282	0.5721	0.6562	0.4849
SLR	precision	0.8261	0.781	0.7353	0.8454	0.7635
	recall	0.84	0.7626	0.7205	0.7457	0.5707
	accuracy	0.9263	0.5384	0.6592	0.211	0.4332
IRQ	precision	0.9681	0.8	0.8247	0.631	0.6503
	recall	0.9555	0.6222	0.7666	0.2407	0.5648



(a) Accuracies

(b) Losses

Figure 4: Grid search in the hyperparameter space. On the x axis, the combinations of timesteps (T) and stride (S) values.

value between consecutive windows, however, does not affect significantly the final score.

We use the best model for each exercise to extract algorithmic coordinates on the test datasets. To compute the segmentation accuracy, we match the algorithmic coordinates against coordinates manually annotated during the data collection phase. Any algorithmic coordinate that matches a manually annotated coordinate within a temporal threshold t_{err} is considered to be a true positive. If an algorithmic coordinate does not have any corresponding ground truth coordinate, it is accounted for as false positive. Lastly, a manually annotated point that is not matched by any algorithmic coordinate is considered to be a false negative. We allow the set of true negative points to be empty (Lin, Karg, and Kulić 2016). A group of baseline techniques is implemented and compared with our method. We generate coordinates using ZVC, HMM, convFSM, and the clustering technique proposed by Bevilacqua et al., hereinafter named *clustree* (Bevilacqua et al. 2018). The segmentation results over the two test sets are summarised in Table 2, for $t_{err} = 35$ (roughly one third of a second) and $n = 25$ (roughly one quarter of a second). These two hyperparameter values were empirically derived before the evaluation process to prevent eccesively distorted segmentation points to be treated as valid coordinates. Our method performs better than all the baseline techniques, for both the healthy and the clinical datasets, showing no sign of oversegmentation.

The motion primitives in the test sets are isolated using the coordinates generated with Algorithm 1 and then classified. In Table 3, we present the classification scores obtained with the boosting strategies described in Section 3. We also include the scores of a number of learning models, namely, Logistic Regression, Random Forest, Support Vector Machines (SVM), Adaptive Boosting (AdaBoost), and J48 Decision Tree. These models are trained using static and dynamic features extracted from the training dataset (Bevilacqua et al. 2018). As the healthy training set contains only instance of the positive class, precision and recall values are superfluous and therefore ignored in Table 3a. It is apparent from this table that our technique yields mixed results, with qDR being generally better than DV and CDS.

6 Conclusions and Future Works

This paper set out to present and evaluate a novel segmentation and classification technique for rehabilitation exercises. Our study was finalized at the resolution of some of the most common issues of the existing methods, such as the coupling with domain knowledge or the inability to work on

Table 3: Test classification score. Best results per exercise are marked in bold.

(a) Healthy set (accuracy)

exercise	DV	MDS	CDS	30DR	50DR	halfDR	Logistic	Rnd. Forest	SVM	AdaBoost	J48
HS	0.9662	**0.9797**	0.9662	**0.9797**	**0.9797**	**0.9797**	0.9391	0.9121	0.831	0.9459	0.9256
SKE	1	**1**	**1**	**1**	**1**	**1**	0.76	0.78	0.78	0.78	0.7733
SLR	0.72	0.7702	0.6466	0.74	0.7333	**0.7933**	0.7929	0.2733	0.6	0.6733	0.5733
IRQ	0.8378	0.7972	0.8783	0.831	0.8445	0.8851	0.8666	0.7933	**0.92**	0.66	0.4466

(b) Clinical set

exercise	metric	DV	MDS	CDS	30DR	50DR	halfDR	Logistic	Rnd. Forest	SVM	AdaBoost	J48
HS	accuracy	0.9968	**1**	0.9968	**1**	**1**	**1**	0.8406	0.8218	0.3156	0.8218	0.8343
	precision	1	1	1	1	1	1	1	1	1	1	1
	recall	0.9968	1	0.9968	1	1	1	0.8406	0.8218	0.3156	0.8218	0.8343
SKE	accuracy	0.6473	0.6431	0.6473	0.6514	0.6473	**0.6514**	0.5311	0.5925	0.6296	0.5791	0.6498
	precision	0.5703	0.5669	0.5725	0.576	0.5714	0.576	0.4647	0.5645	0.6	0.6315	0.6272
	recall	0.7087	0.7	0.6893	0.7	0.7	0.7	0.6407	0.3398	0.3784	0.466	0.67
SLR	accuracy	0.5369	0.5214	0.5369	0.5019	0.4863	0.5291	0.6091	0.5276	**0.6319**	0.6286	0.5309
	precision	0.596	0.5984	0.6043	0.5819	0.5689	0.6046	0.6296	0.5555	0.7397	0.7575	0.5833
	recall	0.6081	0.5135	0.5675	0.4797	0.4459	0.527	0.4594	0.1013	0.3648	0.3378	0.0945
IRQ	accuracy	**0.8843**	0.8097	0.8805	0.832	0.8395	0.8768	0.5862	0.7126	0.7164	0.5632	0.544
	precision	0.9673	0.975	0.9723	0.9647	0.9651	0.9619	0.8125	0.8843	0.7968	0.8303	0.7804
	recall	0.8768	0.7684	0.867	0.8078	0.8177	0.8719	0.5531	0.6914	0.8138	0.4946	0.5106

live data streams. We adopted a mixture of LSTM networks and pointwise boosting techniques to detect cutting points in inertial signals, and produce qualitative classes for the motion primitives of rehabilitation patients. The results of our experimental campaigns suggest that the proposed technique reliably identify edge points in the input signals, and yields classification results comparable to the ones obtained with more traditional learning models, without however relying on any domain knowledge.

Several questions still remain to be answered, and in future works we will investigate (i) the effect of more elaborate boosting techniques, (ii) the adoption of hierarchical classification instead of multiclass classification for the evaluation of the motion primitives, and (iii) the application of our technique on bigger datasets, including heterogeneous exercises and motion patterns.

Acknowledgments

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number 12/RC/2289_P2.

References

American Academy of Orthopaedic Surgeons. 2017. Total knee replacement exercise guide. http://orthoinfo.aaos.org/topic.cfm?topic=a00301. [Online; accessed 4-September-2019].

Argent, R.; Daly, A.; and Caulfield, B. 2018. Patient involvement with home-based exercise programs: Can connected health interventions influence adherence? *JMIR Mhealth Uhealth* 6(3):e47.

Bevilacqua, A.; Huang, B.; Argent, R.; Caulfield, B.; and Kechadi, T. 2018. Automatic classification of knee rehabilitation exercises using a single inertial sensor: A case study. In *2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 21–24.

Bevilacqua, A.; Brennan, L.; Argent, R.; Caulfield, B.; and Kechadi, T. 2019. Rehabilitation exercise segmentation for autonomous biofeedback systems with convfsm. In *IEEE 41st Conference of Engineering in Medicine and Biology*.

Burns, A.; Greene, B. R.; McGrath, M. J.; O'Shea, T. J.; Kuris, B.; Ayer, S. M.; Stroiescu, F.; and Cionca, V. 2010. Shimmer$^{\text{TM}}$ a wireless sensor platform for noninvasive biomedical research. *IEEE Sensors Journal* 10(9):1527 – 1534.

Cheng, P., and Oelmann, B. 2010. Joint-angle measurement using accelerometers and gyroscopes - a survey. *IEEE Trans. Instrumentation and Measurement* 59(2):404–414.

Fod, A.; Matarić, M. J.; and Jenkins, O. C. 2002. Automated derivation of primitives for movement classification. *Autonomous Robots* 12(1):39–54.

Giggins, O. M.; Persson, U. M.; and Caulfield, B. 2013. Biofeedback in rehabilitation. *Journal of NeuroEngineering and Rehabilitation* 10(1):60.

Godfrey, A.; Conway, R.; Meagher, D.; and ÓLaighin, G. 2008. Direct measurement of human movement by accelerometry. *Medical Engineering and Physics* 30(10):1364 – 1386. Special issue for the 30th anniversary of Medical Engineering and Physics.

Jordao, A.; Jr., A. C. N.; de Souza, J. S.; and Schwartz, W. R. 2018. Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art. *CoRR* abs/1806.05226.

Lin, J. F.-S.; Joukov, V.; and Kulić, D. 2018. Classification-based segmentation for rehabilitation exercise monitoring. *Journal of Rehabilitation and Assistive Technologies Engineering* 5:2055668318761523.

Lin, J. F. S.; Karg, M. E.; and Kulić, D. 2016. Movement primitive segmentation for human motion modeling: A framework for analysis. *IEEE Transactions on Human-Machine Systems* 46:325–339.

Tack, C. 2019. Artificial intelligence and machine learning — applications in musculoskeletal physiotherapy. *Musculoskeletal Science and Practice* 39:164 – 169.

Toto, E.; Foley, B. J.; and Rundensteiner, E. A. 2019. Improving emotion detection with sub-clip boosting. In *Machine Learning and Knowledge Discovery in Databases*, 37–52.