Rehabilitation Exercise Segmentation for Autonomous Biofeedback Systems with ConvFSM

Antonio Bevilacqua[1] Louise Brennan[1,2] Rob Argent[1,2] Brian Caulfield[1] Tahar Kechadi[1]

Abstract— **Segmenting physical movements is a key step for any accelerometry-based autonomous biofeedback system oriented to rehabilitation and physiotherapy activities. Fundamentally, this can be reduced to the detection of recurrent patterns, also called motion primitives, in longer inertial signals. Most of the solutions developed in the literature require extensive domain knowledge, or are incapable of scaling to complex motion patterns and new exercises. In this paper, we explore the capabilities of inertial measurement units for the segmentation of upper limb rehabilitation exercises. To do so, we introduce a novel segmentation technique based on Convolutional Neural Networks and Finite State Machines, called ConvFSM. ConvFSM is able to isolate motion primitives from raw streaming data, using very little domain knowledge. We also investigate different combinations of sensors, in order to identify the most effective and flexible setup that could fit a home-based rehabilitation feedback system. Experimental results are presented, based on a dataset obtained from a combination of common upper limb and lower limb exercises.**

I. Introduction

Activity segmentation has a key role in the design and development of autonomous biofeedback systems, especially for home-based rehabilitation supports. Patients that undertake exercise programmes independently, away from a clinical setting, need tools to support them in their rehabilitation. This can include counting individual repetitions, providing granular feedback on the quality of the physical performance, or offering an objective measure of patient adherence [1]. For biofeedback systems based on inertial measures [2], the segmentation step involves the analysis of Inertial Measurement Unit (IMU) data, often sampled as a collection of time series as explained by Wang et al. [3]. In order for a segmentation system to be adopted in real-world scenarios, it is often required to work in a real-time manner, thus the patient does not need to execute the entire set of repetitions before receiving feedback.

The target of this paper is twofold. We inspect the segmentation capabilities of IMU data for a set of upper limb rehabilitation exercises, sampled with three inertial sensors. We also present our exercise segmenter, called ConvFSM. ConvFSM works on streaming signals, and is based on a convolutional classifier and a finite state machine. It relies on minimal domain knowledge, and it is capable of working on raw data, so no data preprocessing or feature engineering is required. In order to test the performance of our system, in addition to the upper limb dataset, we incorporate a further dataset of lower limb exercises for knee rehabilitation.

The rest of this paper is arranged as follows. Section II introduces the problem of motion segmentation, with a brief overview of the most common segmentation approaches developed in literature. A formal definition of the problem is given in Section III, while our proposed approach is detailed in Section IV. Section V contains a short description of our experimental setup and the dataset we used for our test campaigns, for which the results are illustrated in Section VI. Final remarks on the proposed method and suggestions for future developments close the work in Section VII.

II. Related work

The problem of segmenting human motion, and in particular rehabilitation exercises, is a well-known research topic, as it is a key step for most of the autonomous biofeedback systems currently available. In their work, Lin et al. [4] propose a structured taxonomy for segmentation systems, based on many factors such as data sources, application requirements, basic algorithms and validation techniques.

Traditional edge modeling segmentation methods are the zero-velocity crossing (ZVC) techniques [5], based on the detection of points corresponding to a change of movement direction. ZVC points can successfully be extracted from inertial sensor measures, both primitive and derived [6], and they usually do not require detailed domain knowledge except for empirically derived thresholds for edge point grouping. However, segmentation strategies based on ZVC tend to oversegment and do not scale very well with the degrees of freedom (DoFs) of the sampling units [4]. Dynamic Time Warping (DTW) [7] is a popular template-method based on distance measures between signal segments and a motion template. It addresses the different scales and durations that motion primitives can exhibit by computing the best path that warps the observation into the template. This method is proven to be accurate in the segmentation task, however, it is computationally expensive and is designed to work offline, thus it cannot be used in real-world scenarios.

Another common distance-based technique is the adoption of Hidden Markov Models (HMMs) [8] which shape the input signals as a sequence of unobservable Markovian states. HMMs are widely adopted in literature, in a variety of different implementations [9] [10]. However, as for the ZVC methods, they often tend to oversegment, as they do not have a proper rejection mechanism for false positives.

More recent works successfully applied machine learning to the problem of activity segmentation. In their work, Bevilacqua et al. [11] used clustering to find representative

[1] Insight Centre for Data Analytics, University College Dublin, Ireland
[2] Beacon Hospital, Dublin, Ireland

Louise Brennan has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement number 722012 (Cancer: Activating Technology for Connected Health).

points out of all the ZVC points in an exercise. Representative points were later used to extract potential segments, which were classified using a decision tree. This method has the inconvenience of requiring the entire signal to be available before the segmentation process can start. Also, templates are needed to train the decision tree, therefore the model is not easily scalable for novel exercises. Lin et al. [12] based their segmentation process on a binary point classifier capable of discriminating *segment points*, that is, points belonging to a motion segment, from *non-segment points*, i.e., stationary points in between movements. The points are classified using a Support Vector Machine (SVM) trained on the principal components obtained from the data points. Albeit efficient, this segmentation system does not take into account motion patterns where stationary points can be found within exercise segments.

In this paper, we address the most common issues found in the existing segmentation techniques for motion time series, and we apply our system to both lower limb and upper limb exercises, the latter being rather uncommon in research [13].

III. Formal Model

The problem of segmenting physical exercises can be formally described by a set of equations. An exercise E is a set of vector points $\mathbf{p}_t$ collected by the sensor over m time steps as a multivariate time series as outlined in Equation 1.

$$E = < \mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m > \tag{1}$$

Each vector point has as many coordinates as are the degrees of freedom (DoFs) of the sampling device. A single triaxial acceleration point would have three coordinates (a_x, a_y, a_z), while a point obtained from a combination of accelerometer and gyroscope will have six coordinates, $(a_x, a_y, a_z, g_x, g_y, g_z)$. Vector points can also be obtained from multiple sensors, thus the number of components is multiplied by the number of sampling sensors.

The number of motion primitives, also called repetitions, that are included within E is denoted with k. The i^{th} repetition R_i is defined as a subset of points of E, according to Equation 2.

$$R_i \subset E \ , \ \forall i \ : \ 0 \leq i < k - 1 \tag{2}$$

The start and end points of repetition R_i are denoted with s_i and e_i respectively. The repetitions of an exercise do not overlap, so R_i has to be concluded before R_{i+1} can start.

Motion primitives can also include isometric periods of inactivity, varying from subject to subject and from exercise to exercise. We call these points holding points. If the i^{th} repetition R_i was performed as a fluid, individual movement, then the set of holding points $\{h_{i,t}\}$ will be an empty set. Otherwise, $\{h_{i,t}\}$ will denote all the points within repetition R_i that mark the beginning of a change of physical state, from movement to inactivity and vice versa.

Finally, we define the output of the segmentation process for an exercise E as a set of coordinate pairs of length q, as in Equation 3. The points in each pair represent respectively the starting and the ending point of the detected motion primitives in E, and are addressed to as edge points.

$$S(E) = \{(s_j, e_j) \, , \, 0 \leq j < q\} \tag{3}$$

The set of holding points for the exercise E is not specifically a target of a segmentation algorithm.

IV. Proposed approach

Our segmenter is based on the classification of small, overlapping clips, or windows, extracted from the streaming inertial signal. Given the length of the exercise E expressed in Equation 1, the total number of windows that can be extracted from it can be computed with the formula in Equation 4, where W is the window length, P is the padding value, and S is the stride value.

$$w = \frac{m - W + 2P}{S} + 1 \tag{4}$$

As the exercises were not padded, and the stride is set to 1, the total number of windows can be reduced from Equation 4 to $m - W + 1$. Each window can then be classified as either *dormant*, thus corresponding to a period of inactivity, or *dynamic*, thus corresponding to a period of physical motion. This initial classification step is not committed to providing a high-level view of the signal, but it rather identifies regions of movement and regions of inaction. The output of the window classification for the exercise E is a stream of labels, each one corresponding to a window, as indicated in Equation 5.

$$P(E) = < \hat{c}_0, \hat{c}_1, \dots, \hat{c}_w > \tag{5}$$

The predicted window labels are fed into a finite state machine, that is responsible for the generation of a segmentation output like the one illustrated in Equation 3. The FSM models the actual motion pattern that is expected from each exercise, and returns a sequence of edge points for the target exercise. An example is provided in Fig. 1. Here, 2 repetitions of an upper limb exercise are windowed, then segmented. All the windows classified as silence are marked with horizontal arrows, either grey (resting) or red (isometry), while the motion windows correspond to the green regions. The dashed boxes indicate the full extent of the target primitives. In this particular example, the desired output is composed of 4 edge points.

This mechanism presents a number of advantages when compared to traditional techniques. First of all, no feature engineering is required at any stage of the process, as the Convolutional Neural Network (CNN) is able to shape features from the input data on different levels of complexity and nonlinearity, detecting local dependencies over time as well as spatial dependencies over sensors. Also, no templates nor extensive domain knowledge are needed for the segmentation to take place, which makes it easier for our segmentation method to scale to complex motion patterns or simply unseen rehabilitation exercises. Lastly, the proposed approach can be defined as semi-online according to Lin et al. [4], as the training happens offline, while the inference phase natively works on a stream of data.

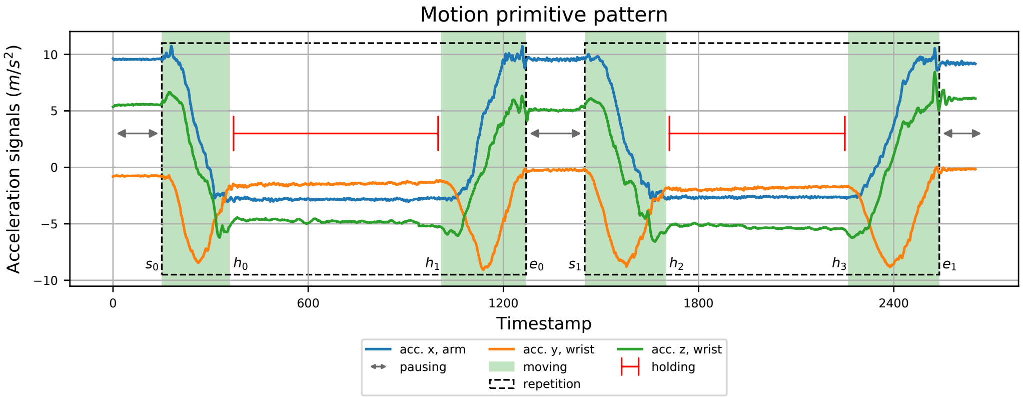

Fig. 1: A pair of repetitions extracted from a shoulder rotation execution, with a holding time of roughly 7 seconds

A. Convolutional classifier

The overlapping windows coming from the streaming signal are fed into a CNN, depicted in the left part of Fig. 2. The input layer accepts image-like windows of shape $6 \times w \times N$, where w is the window length, and N is the number of sensors used for the sampling. This arrangement of the input is known as model-driven [3]. The network architecture is similar to the one developed by Bevilacqua et al. [14], as three convolutional layers and three max pooling layers are followed by a dense network composed of three layers of 500, 250 and 125 units respectively. The kernels for the convolution operation progressively decrease and the kernels for the pooling operation progressively increase. The dense layers are regularized with the dropout technique, with a probability of 0.5 of keeping each neuron. We use a batch size of 2048, for a number of epochs varying from 150 to 300. The initial learning rate for the Adam optimizer is 0.005. The neural network was developed with the TensorFlow framework, version 1.12. The training is executed on a single Nvidia Titan XP GPU.

B. Finite state machine

The FSM models the basic motion pattern of all the target exercises, providing a high-level view of the motion primitives. In principle, an exercise can be represented by a repetitive pattern composed of initial silence, movement, optional silence (corresponding to an isometric contraction) away from the baseline position, movement, then silence again. Most of the target movements in our dataset are isotonic contractions, either concentric or eccentric. For this work, we designed a FSM capable of detecting the basic motion patterns behind our rehabilitation exercises.

One preliminary step behind the working mechanism of the FSM is the baseline test. When an exercise is streamed, baseline values are computed for each signal component, in the form of simple moving average over the same windows that are fed to the CNN. The main assumption behind the baseline computation is that at the very beginning of the exercise, the subjects will hold the starting position for some time, thus providing a reference point on the values the algorithm should expect from each signal component in the resting position. The baseline values are updated whenever new static windows are detected in the pausing state of the FSM. This step is required in order to mitigate the effect of baseline drifting, that is, a gradual displacement from the initial baseline value that occurs when the subject is not fully completing the movement. As shown in Fig. 3, a progressive drift can skew the baseline value substantially.

The FSM model is depicted in the right part of Fig. 2. The state space contains three possible states:

- **moving** (MV): a movement is in progress, either eccentric, concentric, or of noisy nature (vibrations caused by fatigue, pain or distress). This state is reached whenever a window is classified as movement.
- **pausing** (PS): no movement is detected, and the subject limb is close to the initial position. This is the initial state of the FSM, and can be reached from the moving state whenever a window is classified as movement and the baseline test passes.
- **holding** (HL): in this state, no movement is detected, just like for the pausing state, but the exercise is at its isometric peak, as the baseline test failed.

The input symbols of our FSM alphabet are duples composed of the discretized prediction of the current signal window (a value of the sequence in Equation 5), and the outcome of the baseline test. The full alphabet is therefore $\{<\texttt{s},\texttt{true}>,<\texttt{s},\texttt{false}>,<\texttt{m},\texttt{true}>,<\texttt{m},\texttt{false}>\}$, where the values `s` and `m` indicate that the last window was classified as silence or movement respectively, and the values `true` and `false` indicate that the last baseline test passed or failed respectively. For simplicity we include a *don't care* condition for the baseline test, as this value is not always relevant to the state transition, thus reducing the input alphabet to $\{<\texttt{s},\texttt{true}>,<\texttt{s},\texttt{false}>,<\texttt{m},->\}$. If we assign the letters `A`, `B` and `C` to the first, second and third state respectively of the reduced alphabet, we can identify a single exercise repetition with the following sequence of inputs, expressed in form of regular expression: `(A+) (C+) (B (A|B)*)* (C+) (A+)` [1]. The output of the FSM is either a coordinate, or a null value. A

[1] The + symbol indicates one or more of the corresponding value/group, the $*$ symbol indicates zero or more of the corresponding value/group.

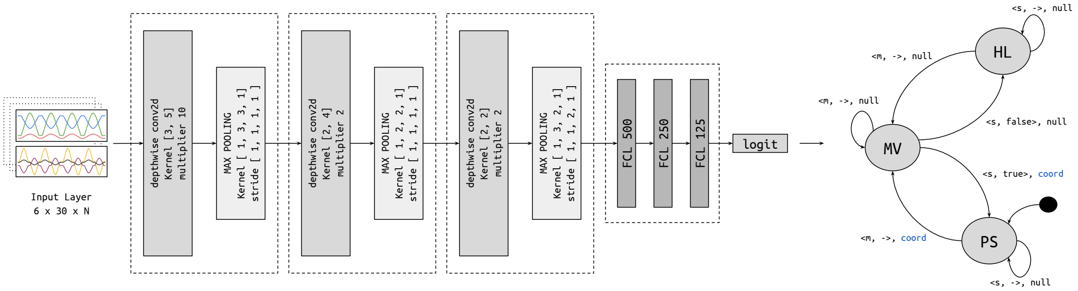

Fig. 2: The architecture of ConvFSM, composed of a CNN for window classification, and a FSM for pattern recognition

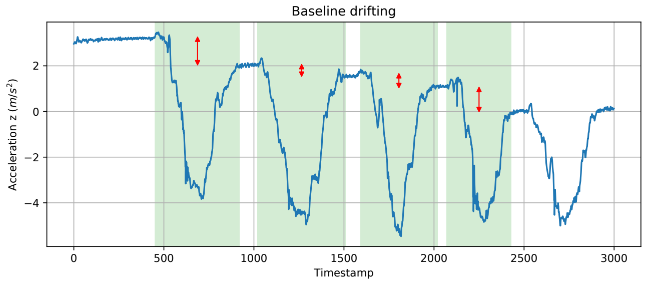

Fig. 3: Baseline drift for an instance of the shoulder abduction exercise sampled with the arm sensor

coordinate value produced in the transition from PS to MV represents the starting point of a primitive, while a coordinate value produced in the transition from MV to PS represents the ending point of a primitive.

V. Data and Methodology

With this paper, we aim at exploring the potential of accelerometry for segmenting upper limb rehabilitation exercises, as well as assessing the performance of our segmentation method. In order to do so, we target different datasets, exercises, and sensor combinations.

A. Dataset composition

Two different datasets are used in this paper, namely, a shoulder dataset, and a knee dataset. The first one is composed of 7 rehabilitation exercises for the upper limb, collected by 29 healthy subjects in a controlled environment. The exercises are shoulder flexion standing (**FLXSTD**), shoulder abduction standing (**ABDSTD**), shoulder internal/external rotation standing (**ROTSTD**), shoulder flexion, active-assisted using a stick (**FLXSTK**), shoulder abduction, active-assisted using a stick (**ABDSTK**), shoulder flexion in supine (**FLXSUP**) and shoulder internal/external rotation in supine (**ROTSUP**). The second dataset is composed of 4 rehabilitation exercises for the lower limb, obtained from a mixed group of 44 clinical subjects and 10 healthy subjects. This dataset was collected and described by [11], and contains samples of heel slide (**HS**), seated knee extension (**SKE**), inner range quadriceps (**IRQ**), and straight leg raise (**SLR**). For both datasets, each participant performed the exercises both in a correct manner and inducing particular deviations instructed by the supervising Chartered Physiotherapist. The deviations collected are elevated scapula, deviation to scapular plane and compensatory trunk extension for FLXSTD, deviation to scapular plane, elevated scapula and compensatory elbow flexion at end of range for ABDSTD, abducting shoulder for ROTSTD, elevated scapula for FLXSTK and ABDSTK, compensatory elbow flexion at end of range for ROTSUP, excessive hip external rotation for HS, lack of full knee extension for SKE, excessive hip flexion for IRQ, inability to maintain full knee flexion for SLR. In addition to the mentioned deviations, FLXSUP and ROTSUP are also executed with 10 seconds of holding time at the isometric peak of the movement.

B. Data acquisition and annotation

All the exercises are sampled with the Shimmer3 [15] inertial unit. This device provides a wide range of sensors, but for this study, we only use the triaxial low noise accelerometer and the triaxial gyroscope, with a fixed sampling rate of 102.4 Hz. The ranges for the accelerometer and the gyroscope are respectively $\pm 2g$ and 500 dps. The shoulder exercises were sampled with three units, placed on the wrist, on the arm, and on the upper trapezius (traps) of the participants, while a single sensor placed on the midpoint of the shin was used to collect the knee dataset.

When acquiring the data for the classification step, the streaming signals are arranged into windows of length 30, corresponding to roughly one-third of a second, as a small window size in most cases can be associated with an improved classification accuracy [16]. Consecutive windows overlap with a stride of 1 point. The accelerometer and the gyroscope are both triaxial, so each input example comes in the shape $6 \times 30 \times N$, where N represents the number of sensors used by the participant during the exercise. Whilst only one sensor is available for the knee dataset, different combinations of the three sensors were tested for the shoulder dataset: individual sensor setups, double sensor setups (arm and wrist, arm and traps, wrist and traps), and the combination of all three sensors at once.

After the data acquisition, each exercise is manually annotated. During this phase, both edge points and hold points are accounted for. This is necessary, as the ground labels for the windows are assigned so that only real movement is marked as such. In the example signal showed in Fig. 1, the entire set of points $\{s_0, h_0, h_1, e_0, s_1, h_2, h_3, e_1\}$ is annotated, so that all the windows between the regions $[s_0, h_0]$, $[h_1, e_0]$, $[s_1, h_2]$ and $[h_3, e_1]$ can be labeled as movement, while the regions

$[0, s_0]$, $[h_0, h_1]$, $[e_0, s_1]$, $[h_2, h_3]$ and $[e_1, m]$ are labeled as silence. However, during the evaluation phase only the edge points $\{s_0, e_0, s_1, e_1\}$ are considered as ground edge points.

VI. Experimental results

For our experimental campaign, we first present the bare classification accuracy obtained for the window classification task, then we compute the counting accuracy, for all the sensor configurations in the case of the upper limb dataset, and for the knee sensor in the case of the lower limb dataset. All the metrics are computed with the *Leave-One-Subject-Out* strategy [17], useful to induce intersubject variability in the experimental setup. The final results are then obtained as an average of all the individual folds, where the test set of each fold is composed by a subject that has not been seen during the training phase. Each entry also includes the standard deviation across all the folds. We also compare our segmentation scores with the scores obtained from two of the most common existing techniques, namely, HMM and ZVC. For the first one, we train a model for each exercise in the dataset with the Baum-Welch algorithm, then we use the forward algorithm to compute the similarity with new observation data. As for the second approach, we compute ZVC points on each exercise execution, by leveraging only the most significant signal components, in order to alleviate the oversegment effect of this technique.

When computing the segmentation accuracy, we count an algorithmic edge point as true positive if it matches a manually annotated edge point within an interval of 100 time steps, roughly corresponding to $0.5s$ on each shift direction. If such a mach does not exist, the algorithmic edge point is counted as false positive. All the manually annotated edge points that do not have a match among the algorithmic edge points are counted as false negative. This validation metric admits the true negative points to be an empty set [4].

Table I reports the window classification accuracy, loss, precision and recall for every sensor and sensor combination, for both the shoulder and then knee datasets. For the shoulder dataset, the poorest result is obtained with the traps sensor, while both the arm and the wrist sensors yield an accuracy of at least 87% when used alone. Combining the arm and the wrist sensors with the traps sensors does not increase the overall accuracy, whilst the combination of wrist and arm sensors increases the accuracy with respect to the single sensors by 0.2% and 1.6% respectively. The highest overall accuracy is obtained by using all three sensors together.

The segmentation scores of the upper limb exercises obtained with ConvFSM are listed in Table II. The accuracy scored when counting motion primitives is generally higher than the bare classification accuracy over individual windows. This is due to the fact that the FSM has some tolerance towards misclassified windows, as coordinates of motion primitives cannot be generated while in holding. It is possible to notice that the worst-performing sensor is traps, and that the combination of all the three sensors often returns better accuracy than single sensor configurations. The 2 exercises executed while holding a stick, FLXSTK and ABDSTK, seem to be relatively easy to segment when compared to their standing versions. Holding a stick in both hands during the exercise stabilises the upper limb of the subjects, likely causing lower variation across the IMU signals.

Table IIIa displays a comparison of ConvFSM with HMM and ZVC. For sake of brevity, we only include the accuracy obtained with the best sensor for each algorithm. The tested methods work well for some of the exercises, however, they still show a tendency to oversegment, as precision scores are often low. The wrist sensor appears to be the best placed sensor for segmenting the target activities, as in the majority of the experiments, it provides the best segmentation performance. In general, ConvFSM outperforms HMM and ZVC, as it is not penalized by excessive edge point production.

Table IIIb reports the segmentation accuracy for the knee dataset, side by side with the segmentation accuracy values obtained with ZVC and HMM. Also in this case, ConvFSM reaches better results than the two baseline models.

VII. Conclusions and future works

In this paper, we presented ConvFSM, a segmentation method for multimodal time series based on a convolutional classifier and a finite state machine. We applied ConvFSM to the segmentation of rehabilitation exercises sampled with IMUs, and we were successful in detecting individual movements over two datasets of common lower and upper limb rehabilitation exercises. ConvFSM can natively be used on a streaming data source, regardless of the number of sensors used for the sampling, and does not require extensive domain knowledge about the target exercises. It also works on raw data, so it does not require data preprocessing, data cleaning, nor feature engineering. Our results show that ConvFSM performs reasonably good on a wide number of exercises, for both the lower part and the upper part of the body.

Despite its flexibility, ConvFSM presents a few limitations. The main assumption behind the FSM implementation is that the very beginning of the streaming is composed of inactivity windows, so that the initial baselines can be computed. In case the stream opens with motion windows, ConvFSM may return unexpected results. This problem can be solved with the introduction of *suggested* baseline values, however, prior knowledge would be required about the orientation of the used sensors. Also, ConvFSM tends to include in the primitives small segments at the beginning and the end of the movement that reflect limb vibration. These segments would not ideally add any benefit for the segment classification, so further adjustment of the extracted primitives could be required before they could be properly classified.

In the future, we aim at testing the segments obtained with ConvFSM for movement classification, in order to understand the real impact different coordinates may have on the final feedback provided to the patients. Also, we want to extend our datasets, to include more exercises and participants. As a final remark, the generation of a single ConvFSM system that can be used for both upper limb and lower limb movements is worth further investigation.

TABLE I: Window classification scores

metric	**WRIST**	**ARM**	**TRAPS**	**W, A**	**W, T**	**A, T**	**W, A, T**	**KNEE**
accuracy	0.949 ± 0.01	0.935 ± 0.017	0.869 ± 0.027	0.951 ± 0.016	0.9488 ± 0.02	0.9336 ± 0.0156	0.9521 ± 0.01	0.9382 ± 0.0242
loss	0.16 ± 0.07	0.18 ± 0.06	0.3 ± 0.049	0.14 ± 0.06	0.14 ± 0.0384	0.18 ± 0.0385	0.14 ± 0.0436	0.2 ± 0.1781
precision	0.968 ± 0.014	0.96 ± 0.021	0.92 ± 0.035	0.974 ± 0.021	0.972 ± 0.0126	0.9684 ± 0.018	0.975 ± 0.0162	0.961 ± 0.029
recall	0.958 ± 0.018	0.945 ± 0.023	0.89 ± 0.048	0.9554 ± 0.0158	0.954 ± 0.0182	0.9348 ± 0.023	0.9553 ± 0.0154	0.9448 ± 0.0301

TABLE II: Segmentation scores, shoulder dataset

exercise	**metric**	**WRIST**	**ARM**	**TRAPS**	**W, A**	**W, T**	**A, T**	**W, A, T**
FLXSTD (2072)	accuracy	0.9913	0.975	0.91	0.999	0.9951	0.9971	0.999
	precision	0.998	0.9946	0.9621	1	0.9966	1	1
	recall	0.9932	0.9802	0.9444	0.999	0.99985	0.9971	0.999
ABDSTD (2158)	accuracy	0.9981	0.9953	0.9247	0.9981	0.9916	0.9944	0.9981
	precision	0.999	0.9986	0.9534	1	0.9981	0.9972	0.9986
	recall	0.999	0.9967	0.9684	0.9981	0.9935	0.9972	0.9995
ROTSTD (1050)	accuracy	0.7228	0.6913	0.5051	0.8	0.756	0.7026	0.8086
	precision	0.9079	0.743	0.648	0.894	0.8868	0.7195	0.8793
	recall	0.78	0.9085	0.6961	0.8838	0.8361	0.9676	0.9095
FLXSTK (1052)	accuracy	0.9943	1	0.8641	1	0.9641	0.9792	1
	precision	0.998	1	0.9352	1	0.9912	0.99	1
	recall	0.996	1	0.9192	1	0.9724	0.9885	1
ABDSTK (1036)	accuracy	0.9846	0.998	0.9492	0.998	0.99	0.99	0.99
	precision	0.9932	1	0.974	0.999	0.997	0.9951	0.9952
	recall	0.9913	0.998	0.974	0.999	0.9932	0.9951	0.9952
FLXSUP (898)	accuracy	0.9561	0.926	0.6245	0.9694	0.8974	0.948	0.9693
	precision	0.9988	0.9918	0.778	0.9933	0.9939	0.9886	0.9988
	recall	0.9572	0.9331	0.7587	0.9758	0.9024	0.9583	0.9704
ROTSUP (898)	accuracy	0.9698	0.7481	0.496	0.8412	0.99	0.6851	0.9484
	precision	0.999	0.9065	0.6144	1	1	0.8655	0.9556
	recall	0.97	0.8107	0.72	0.8412	0.99	0.7666	0.992

TABLE III: Segmentation scores obtained from best individual sensor systems, shoulder dataset

algorithm	**metric**	**FLXSTD**	**ABDSTD**	**ROTSTD**	**FLXSTK**	**ABDSTK**	**FLXSUP**	**ROTSUP**
ZVC	sensor	wrist	wrist	wrist	wrist	wrist	wrist	wrist
	accuracy	0.9326	0.8869	0.2314	0.8713	0.8928	0.6126	0.7652
	precision	0.9476	0.8958	0.2622	0.8728	0.8967	0.8153	0.9587
	recall	0.9777	0.9888	0.6638	0.998	0.9951	0.7114	0.7912
HMM	sensor	arm	arm	wrist	wrist	arm	wrist	wrist
	accuracy	0.9866	0.9104	0.4573	0.9495	0.916	0.6785	0.7868
	precision	0.99	0.9112	0.5688	0.9495	0.9267	0.9824	1
	recall	0.9956	0.999	0.7	1	0.9874	0.6868	0.7868
ConvFSM	sensor	wrist	wrist	wrist	arm	arm	wrist	wrist
	accuracy	0.9913	0.9981	0.7228	1	0.998	0.9561	0.9698
	precision	0.998	0.999	0.9079	1	1	0.9988	0.999
	recall	0.9932	0.999	0.78	1	0.998	0.9572	0.97

(a) Best individual sensor systems, shoulder dataset

algorithm	**metric**	**HS**	**SKE**	**IRQ**	**SLR**
ZVC	accuracy	0.1823	0.579	0.5908	0.5882
	precision	0.1972	0.6646	0.6738	0.5906
	recall	0.7072	0.818	0.8275	0.9931
HMM	accuracy	0.6534	0.742	0.345	0.4136
	precision	0.6625	0.765	0.4767	0.4147
	recall	0.9793	0.96	0.5553	0.994
ConvFSM	accuracy	0.9225	0.9554	0.8807	0.8798
	precision	0.9321	0.9975	0.9885	0.9263
	recall	0.989	0.9576	0.8892	0.946

(b) Individual sensor systems, knee dataset

References

[1] R. Argent, A. Daly, and B. Caulfield, "Patient involvement with home-based exercise programs: Can connected health interventions influence adherence?" *JMIR Mhealth Uhealth*, vol. 6, no. 3, p. e47, Mar 2018. [Online]. Available: https://mhealth.jmir.org/2018/3/e47/

[2] O. M. Giggins, K. T. Sweeney, and B. Caulfield, "Rehabilitation exercise assessment using inertial sensors: a cross-sectional analytical study," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 158, Nov 2014. [Online]. Available: https://doi.org/10.1186/1743-0003-11-158

[3] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *CoRR*, vol. abs/1707.03502, 2017. [Online]. Available: http://arxiv.org/abs/1707.03502

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

[5] A. Fod, M. Mataric, and O. Jenkins, "Automated derivation of primitives for movement classification," vol. 12, 06 2003.

[6] A. Godfrey, R. Conway, D. Meagher, and G. Laighin, "Direct measurement of human movement by accelerometry," 01 2009.

[7] M. Müller, *Dynamic Time Warping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 69–84. [Online]. Available: https://doi.org/10.1007/978-3-540-74048-3_4

[8] Y. Bengio, "Markovian models for sequential data," 1999.

[9] J. F. S. Lin and D. Kulić, "Automatic human motion segmentation and identification using feature guided hmm for physical rehabilitation exercises," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Workshop on Robotics for Neurology and Rehabilitation*, San Francisco, CA, 2011, p. 33–36.

[10] J. Lin and D. Kulic, "Segmenting human motion for automated rehabilitation exercise analysis," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2012, pp. 2881–4, 08 2012.

[11] A. Bevilacqua, B. Q. Huang, R. Argent, B. Caulfield, and M. T. Kechadi, "Automatic classification of knee rehabilitation exercises using a single inertial sensor: A case study," pp. 21–24, 2018. [Online]. Available: https://doi.org/10.1109/BSN.2018.8329649

[12] J. F.-S. Lin, V. Joukov, and D. Kuli, "Classification-based segmentation for rehabilitation exercise monitoring," *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 5, p. 2055668318761523, 2018. [Online]. Available: https://doi.org/10.1177/2055668318761523

[13] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," *CoRR*, vol. abs/1611.06455, 2016. [Online]. Available: http://arxiv.org/abs/1611.06455

[14] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and T. Kechadi, "Human activity recognition with convolutional neural networks," in *Machine Learning and Knowledge Discovery in Databases*, U. Brefeld, E. Curry, E. Daly, B. MacNamee, A. Marascu, F. Pinelli, M. Berlingerio, and N. Hurley, Eds. Cham: Springer International Publishing, 2019, pp. 541–552.

[15] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "Shimmer$^{\text{TM}}$ a wireless sensor platform for noninvasive biomedical research," *IEEE Sensors Journal*, vol. 10, no. 9, pp. 1527 – 1534, 2010.

[16] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Evaluating the effects of signal segmentation on activity recognition," in *International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO 2014*, 2014, pp. 759–765.

[17] A. Jordao, A. C. N. Jr., J. S. de Souza, and W. R. Schwartz, "Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art," *CoRR*, vol. abs/1806.05226, 2018. [Online]. Available: http://arxiv.org/abs/1806.05226