

Article

An Online Method for Supporting and Monitoring Repetitive Physical Activities Based on Restricted Boltzmann Machines

Marcio Alencar ^{*}, Raimundo Barreto , Eduardo Souto  and Horacio Oliveira 

Institute of Computing, Federal University of Amazonas, Manaus 1200, Brazil;

rbarreto@icomp.ufam.edu.br (R.B.); esouto@icomp.ufam.edu.br (E.S.); horacio@icomp.ufam.edu.br (H.O.)

* Correspondence: macalencar@icomp.ufam.edu.br

Abstract: Human activity recognition has been widely used to monitor users during physical activities. By embedding a pre-trained model into wearable devices with an inertial measurement unit, it is possible to identify the activity being executed, count steps and activity duration time, and even predict when the user should hydrate himself. Despite these interesting applications, these approaches are limited by a set of pre-trained activities, making them unable to learn new human activities. In this paper, we introduce a novel approach for generating runtime models to give the users feedback that helps them to correctly perform repetitive physical activities. To perform a distributed analysis, the methodology focuses on applying the proposed method to each specific body segment. The method adopts the Restricted Boltzmann Machine to learn the patterns of repetitive physical activities and, at the same time, provides suggestions for adjustments if the repetition is not consistent with the model. The learning and the suggestions are both based on inertial measurement data mainly considering movement acceleration and amplitude. The results show that by applying the model's suggestions to the evaluation data, the adjusted output was up to 3.68x more similar to the expected movement than the original data.

Keywords: physical activities; pattern recognition; restricted boltzmann machine; movement adjustments



Citation: Alencar, M.; Barreto, R.; Souto, E.; Oliveira, H. An Online Method for Supporting and Monitoring Repetitive Physical Activities Based on Restricted Boltzmann Models. *J. Sens. Actuator Netw.* **2023**, *12*, 70. <https://doi.org/10.3390/jsan12050070>

Academic Editor: Lei Shu

Received: 20 May 2023

Revised: 13 September 2023

Accepted: 14 September 2023

Published: 22 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The movement recognition, usually identified as a typical pattern recognition problem and, more specifically, a classification problem [1], has been applied in sports by using machine learning techniques to generate neural network models for multiple purposes. These purposes include: (i) identifying critical conditions (e.g., falls, heart attacks); (ii) classifying activities (e.g., swimming, running, cycling); and (iii) counting repetitions (e.g., steps, jumps, squats) [2–4]. The importance of analyzing physical activities with the help of neural networks models and wearable devices lies in the ability to derive insightful information from these models that can be relevant to the user's performance. For example, predicting sweat loss to prevent dehydration based on factors such as heart rate, temperature, anthropometric parameters of users, and steps per minute, as presented in [5].

In the context of repetitive physical activities, to the best of our knowledge, the lack of a neural network model to provide qualitative feedback that helps users to improve, or adjust, the movement for proper execution is still an open problem. Commonly performed in gyms or physiotherapy sessions, the repetitive physical activities focus on strengthening a set of muscles by activating them along multiple sessions for several months or years. However, overstressing these muscles by incorrect movement executions may cause ruptures and lesions that demand rest or, in the worst case, surgical intervention to reconstruct the muscle fiber. By developing a technology to assist the users regarding their movement execution, we can drastically change the relevance of wearable devices/neural networks from a simple monitoring technology to a disruptive approach that improves the users' performance and prevents injuries [6].

At this point, we can say that our research question is about data reconstruction. It involves fixing incorrect input data, which represents a wrong execution of physical activity, to create accurate output data that shows the right way of doing it. In simpler terms, we are exploring whether it is possible to correct input data and provide feedback that helps users improve their future input data.

To address the research question effectively, we must also answer the following motivating questions: (i) is it possible to provide a neural network model that is capable of generating a qualitative feedback to the users during repetitive physical activities (e.g., squats, chest fly, row, curl)? (ii) is it possible to improve the users' performance on repetitive physical activities by adjusting the movements from the model's feedback? and (iii) is it possible to generate a model that adapts to the user's limitations avoiding misleading generalizations that are not suitable for the user's physical traits?

This paper proposes an alternative solution to face the challenges and difficulties, advocating that it is possible, at runtime, to generate both the appropriate pattern and the model for a specific activity, from the use of inertial sensors along the various body segments. Each generated model can provide detailed adjustment suggestions that guide the users to properly perform a physical activity. This approach avoids misleading generalizations by generating a new model at the beginning of each different activity, identifying the movement pattern most suitable for the user's physical traits at that time, respecting the limitations of the users according to their progress, preventing overstress the muscles and keeping the movement quality along the physical activity sets.

Despite the various approaches to convert, or map, the movements made by individuals in the real world into data that can be analyzed, such as using cameras [7–9], electromyography [10–13], and resistive sensors based [14–17], this paper considers the inertial sensors as the best approach due its cost/benefit. These sensors have a small size, good accuracy, and low cost, in addition to being able to measure the monitored segment orientation based on linear acceleration (accelerometer) and angular velocity (gyroscope) in three different axes (x , y and z) [18,19].

The proposed method was evaluated using the PHYTMO—Physical Therapy Monitoring [20] dataset which contains inertial sensors data from the body's segments classified as correct and incorrect execution, allowing to compare the proposed method's efficiency. These data were analyzed by two main algorithms: (i) Dynamic Time Warping (DTW) [21,22] and (ii) The Restricted Boltzmann Machine (RBM) [23]. The RBM is used to extract the patterns from the training data and to generate the adjustment suggestions for new input data, and the DTW is used to evaluate the efficiency of the proposed method by identifying the gain in applying the adjustment made by the RBM model in the same input data.

The remainder of this paper is organized as follows. The next Section presents the Related Works, followed by Theoretical Background which presents the base concepts needed to understand this paper. The Proposed Method Section explains, in multiple subsections, an alternative to address the problems mentioned before. The Evaluation Method describes the steps to validate the proposed method and describes the obtained results and its discussions. Finally, the Conclusion Section presents an overview, achievements, limitations of the proposed method and perspectives for future work.

2. Related Works

The most similar study found in the literature proposes the use of a Recurrent Restricted Boltzmann Machine to perform predictions of chaotic time-series [24]. Despite the divergence of the application, the proposed method would be an interesting candidate to solve the research problem in this manuscript. Their proposes includes a recurrent structure at the hidden nodes that stores historical information, generating more accurate predictions for future values for the input time series.

The proposed method in [25] uses a support vector machine (SVM) classifier to recognize postural patterns through wearable sensors to avoid postures that increase spinal

stress. This approach trained a model to recognize lifting and releasing movements with correct and incorrect postures using kinematic data from 8 sensors distributed on the lower and higher legs and in the trunk body segment. The experiments involved 26 healthy subjects, and the SVM model achieved an accuracy of 99.4% in identifying, at real-time, correct and incorrect postures, considering all sensors. Despite the good results, this model only works for those movements (lift and release loads). Furthermore, SVM only generates outputs that have already been evaluated during the training process, unlike RBM which can adapt and generate outputs based on a previous distribution of the training dataset.

The studies conducted in [26] provide a deep learning framework capable of evaluating up to 10 rehabilitation exercises. It uses an autoencoder as sub-networks to process the displacement of individual body parts, which are monitored by a visual motion sensor. Their conclusions demonstrate that probabilistic models outperform approaches that use distance functions for movement assessments. A second outcome of this research is a dataset called UI-PRMD with data collected from 10 healthy subjects, composed by 10 repetitions of 10 rehabilitation exercises. This data were gather by an optical tracking system and consist of 117-dimensional sequences of angular joint displacements.

A different reconstruction time series approach, proposed in [27], uses a denoising autoencoder to reconstruction time series with missing values. By converting the raw time series into a 2d matrix that establishes correlations between the time intervals, it is possible to use a Denoising autoencoder to reconstruct the missing value in the 2D matrix. The results show that uses 2D representations of a time series improves the imputation and classification performance.

The study in [28] evaluates four machine learning techniques regarding the binary classification of an exercise. It reached a misclassification error of up to 0.5%, using a support vector machine with a polynomial kernel, and up to 99% of accuracy in detecting wrong movements considering 7 different exercises on physical therapy routines. As explained previously, this approach also has a limited number of exercises that it is capable of recognizing and classifying, reducing its applications due to difficulties in adding and removing different routines. Also, the authors states that wearable device cannot detect a variety of fitness movements and may hinder the exercises of the fitness users.

An interesting approach in [29] proposes the use of transferring learning based on deep neural networks to increase the number of classes (physical activities) that could be evaluated by the model. The performance experiments reached up to 98.56% of accuracy and 97.9% precision in identifying the movements. In the movement completeness, it reached up to 92.84% of accuracy and 92.85% precision. Despite the good results, the transfer has a previous cost of generating a generic model. Furthermore, scalability and necessary model updates would be difficult to maintain over time.

In the literature, there is no paper able to satisfy the main question that motivates our study: how can we propose an approach to generate specific models, in runtime, to assist each user based on their physical characteristics to perform a physical activity correctly given them adjustment suggestions?

3. Theoretical Background

This Section introduces the basic concepts for understanding the proposed method, starting with the Inertial Sensors Data Section, which explains how to combine the raw sensor's data into a single metric to reduce the dimensionality simplifying the data analysis. Moreover, it explains the RBM algorithm, how it works and why it is the chosen algorithm to integrate the proposed method. Finally, it is presented the Dynamic Time Warping algorithm which is used to evaluate the gain on applying the RBM suggestion into the original input series.

3.1. Inertial Sensor Data

Despite the various approaches to convert, or map, the movements made by individuals in the real world into data that can be analyzed, such as by using cameras [7–9],

electromyography [13,30], and based on resistive sensors [14,31], this paper considers that inertial sensors are suitable candidates for that. These sensors have a small size, good accuracy, and low cost, in addition to being able to measure the monitored segment orientation based on linear acceleration (accelerometer) and angular velocity (gyroscope) in three different axes (x , y and z) [18,19,32].

The mapping process consists of registering multiple readings during the movement execution. This set of readings, called time series, contains raw data from three axes for each sensor that helps to understand the behavior of a monitored segment regarding the acceleration and the angular velocity. The raw time series data are combined to reduce the data dimensionality by using the magnitude metric [33,34]. This metric is $M_{sens} = \sqrt{x_t^2 + y_t^2 + z_t^2}$, where x_t , y_t , and z_t are the measurements on each axis by the sensor ($sens$) at a specific instant (t) during the movement execution.

Finally, the accelerometer and the gyroscope magnitude values are combined with a balanced fusion operation given by $F_t = (0.5 \times M_{acc_t}) + (0.5 \times M_{gyr_t})$, where M_{acc_t} and M_{gyr_t} are the magnitude values from the accelerometer and gyroscope at instant t , respectively. Figure 1 presents a sample of raw data from the accelerometer and gyroscope, and their conversion to the magnitude metric and the data fusion.

The advantage of using the magnitude metric and the fusion of magnitudes relies on the conversion of negative values into positive values, and the amplification of the wave, as depicted in Figure 1: Magnitude–Fusion (orange line). These characteristics help to highlight the cyclical pattern, implicit in the raw data of each sensor’s axis and, at the same time, they reduce the data dimensionality, simplifying the data analysis by using the absolute values. Also, problems with sensor displacement and axis orientations are mitigated by this metric because it considers only the absolute values of the forces in all of the axes of the sensors [33].

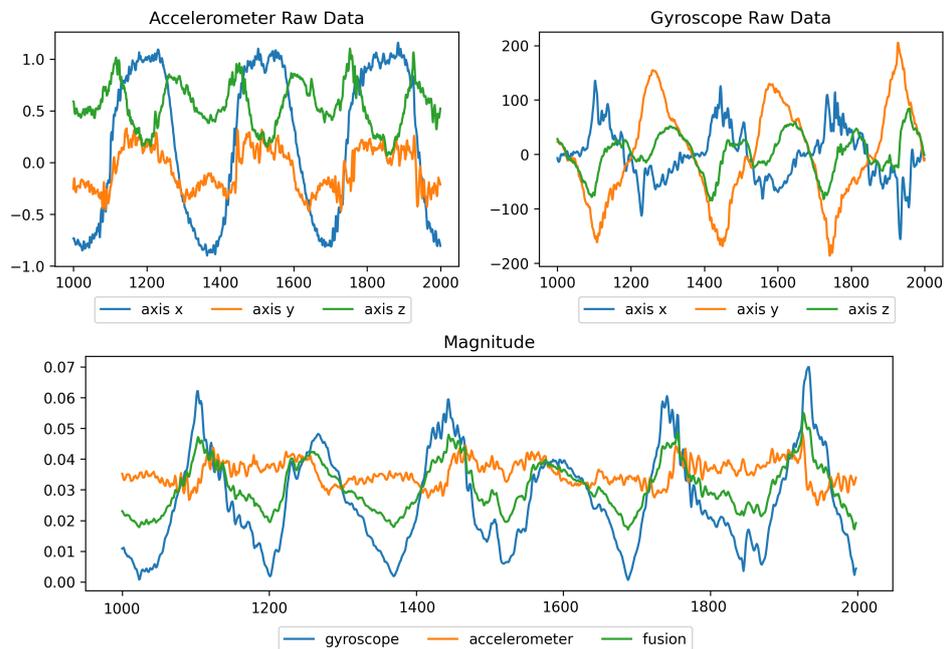


Figure 1. Accelerometer (raw), gyroscope (raw) and magnitudes (fusion).

3.2. Restricted Boltzmann Machines

The Restricted Boltzmann Machine (RBM) [23] is a two-layer (visible and hidden) artificial neural network with no within-layer units’ connections that make stochastic decisions about whether units should be on (activated) or off (deactivated). This network receives a set of binary arrays and learns the set pattern (adjust the weights between the units— w_{ij}) by finding a prior distribution in the observed data to generate arrays with high probability, as depicted in Figure 2. It uses Gibbs sampling during this process.

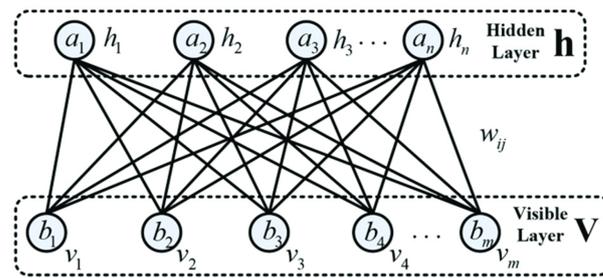


Figure 2. Visual representation of a Restricted Boltzmann Machine. Adapted from [35].

Differently from the Feed Forward Neural Networks [36], the RBM iterates over the layers, activating and deactivating the units until the visible layer represents a set that satisfies the previous distribution data. This characteristic demands high-quality input data during the training process, otherwise the weights would "learn" bad patterns and generate undesired visible units' activation.

This neural network was chosen by its common application on recommendation systems [37,38]; also, its stochastic nature allows to identify a pattern from multiple subsets of a probabilistic distribution that would represent the training dataset; and, finally, this neural network contains only two layers, being suitable to embed on constrained resource devices.

3.3. Dynamic Time Warping

Dynamic Time Warping (DTW) [21,22] is applied to compute the distance between two time series by finding the best alignment between them. It uses dynamic programming to combine the points from both series to extract the minimal combination cost that expresses the total distance between both series. The smaller the total distance is, the more similar the compared time series are. This approach outperforms the traditional Euclidean Distance since it compensates for eventual differences in the frequency and duration of both series.

Figure 3 illustrates the best alignment path (connected dots on the matrix) between two time series (x and y) by associating the closest elements to each other (arrows connecting the elements' arrays) on both series.

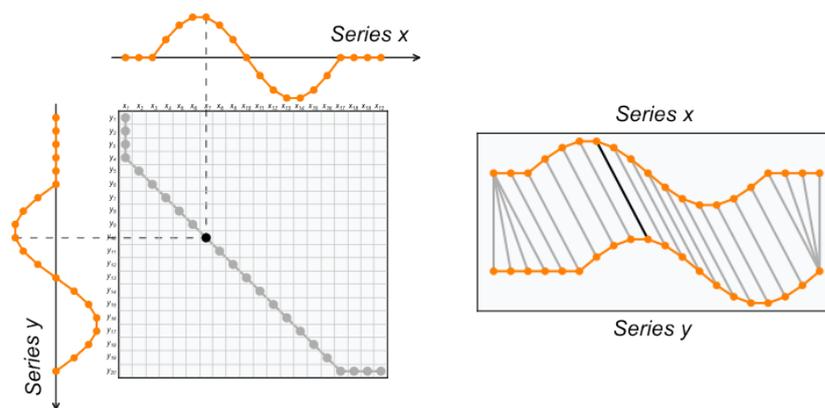


Figure 3. The best alignment between both series (x and y). Adapted from [39].

This algorithm was originally created to perform voice recognition, where the same word could be spoken with a different tone, frequency, and speed. By compensating for these variables, it is possible to adjust both series to identify relevant patterns between them.

Applying this technique to compare two magnitude series helps to identify the similarity between them. Using a reference series, it is possible to calculate the gain on applying the RBM suggestions in the input series by comparing both (input and changed series) to the reference model and calculating the difference between the results. Alternatively, it

is used to perform a filtering on training samples that helps to generate a more accurate RBM model.

4. Proposed Method

Once the theoretical background provided the basic knowledge to understand the proposed method, this Section explains the pre-processing step (Series Binarization Section) that prepares the magnitude time series to train the RBM model and to analyze the further series by converting them into a binary array. This Section also describes the RBM architecture, specifying how the number of visible and hidden nodes are defined. After that, the proposed method pipeline is described for both training and the evaluation of new movements.

4.1. Series Binarization

The proposed method states that the difference between the sample values at the time $t + 1$ and t is represented as a 3-tuple of discrete states which indicate if the value is decreasing (1,0,0), sustaining (0,1,0), or increasing (0,0,1). This means that a sample (a single execution of the physical activity) containing n values of magnitude readings is represented by a binary array of $3 \times (n - 1)$ elements (or $n - 1$ 3-tuples).

For example, assuming a time series of magnitude values $x = [0.45, 0.53, 0.69, 0.72, 0.7, 0.7, 0.68, 0.65\dots]$, the binarization considers the variation between x_t and $x_{t + 1}$ to generate the 3-tuple. In this case, for the first ($x_0 = 0.45$) and the second ($x_1 = 0.53$) elements of x , we have $0.53 - 0.45 = 0.08$, which is a positive number ($0.08 > 0$) that indicates that between x_0 and x_1 , the magnitude is increasing; then, the 3-tuple representation would be (0,0,1). Following these steps for the further elements in x , the resultant array of 3-tuples is $x_{3-tuples} = [(0,0,1), (0,0,1), (0,0,1), (1,0,0), (0,1,0), (1,0,0), (1,0,0)\dots]$. Despite each element corresponding to a 3-tuple, for the RBM, all tuples are combined sequentially, forming an array of binary values holding the position of each element as follows: $x_{bin} = [0,0,1,0,0,1,0,0,1,1,0,0,0,1,0,1,0,0,1,0,0\dots]$. This binary array translates the tendencies of values in the magnitude time series which will be placed as input values in the RBM for both steps, training and evaluating.

4.2. The RBM Architecture

The visible layer of the RBM must handle all elements presented by the binary array, generated by the Series Binarization; this means that the RBM architecture will have a visible layer composed of $3 \times (n - 1)$ units, with each one of them representing a single element from a binary array.

The hidden layer is composed of a set of $n - 1$ hidden units, representing the number of 3-tuples in the visible layer. This architecture setup assumes that a trained model activates/deactivates the unit from the 3-tuples to satisfy the most probable states according to the model weights and its hidden units. The changes made by the model are the adjustment recommendations that transform the "incorrections on the input series" (original visible units) into a "corrected output series" (changed visible units).

The output 3-tuples represent the expected tendency of the sensor's readings along the movement. For example, if the model "learns" that between the readings at the time t_i and $t_i + 1$, the value must increase, then it changes the input data to represent this tendency.

As presented with the pre-processing step and the RBM architecture, it is possible to present the Pipeline Overview Section, which uses all base concepts presented before in this manuscript.

4.3. Pipeline Overview

The proposed method states that it is possible to provide feedback by generating specific models, in execution time, for each individual body segment, based on fusion magnitude time series (presented in the Inertial Sensor Data Section) from inertial sensors.

Figure 4 depicts the training process pipeline which starts by collecting a set of high-quality data during the right movement execution, it is possible to highlight the cyclic patterns, or periods (see Figure 1), where each one of them represents an exercise repetition (or sample). These samples are converted into a binary array (presented in the Section 4.1) placed as visible units from an RBM model (see Figure 2) so it could be possible to train the model.

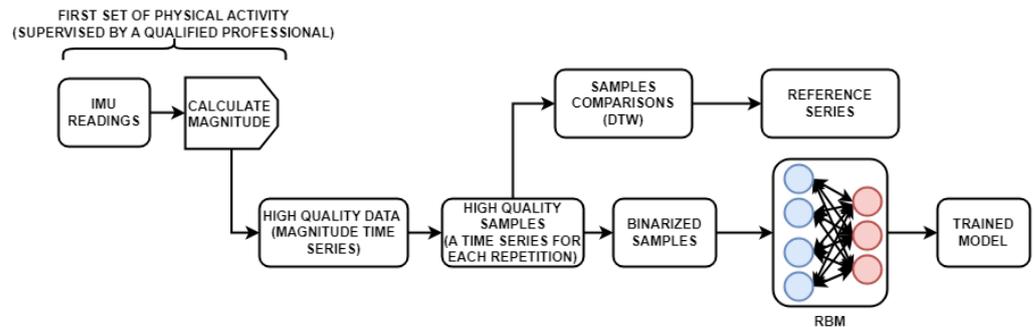


Figure 4. Training process of the proposed method.

For each different physical activity, the model must be retrained to replace the outdated model and to adapt to the user’s rhythm along their evolution on the physical activities’ practices, avoiding misleading generalizations from massive datasets. Also, the evaluation of independent segments by specific models would provide accurate feedback, preventing interference from the other sensors on the pattern’s extractions and analysis.

Once the model is generated, then is possible to evaluate the following set of repetitive physical activities obtaining the samples from input data, converting them into a binary array and then feeding the model, which, based on a prior distribution from the training dataset, will reconstruct/modify the input data to satisfy the patterns from the correct movement execution as illustrated in the Figure 5.

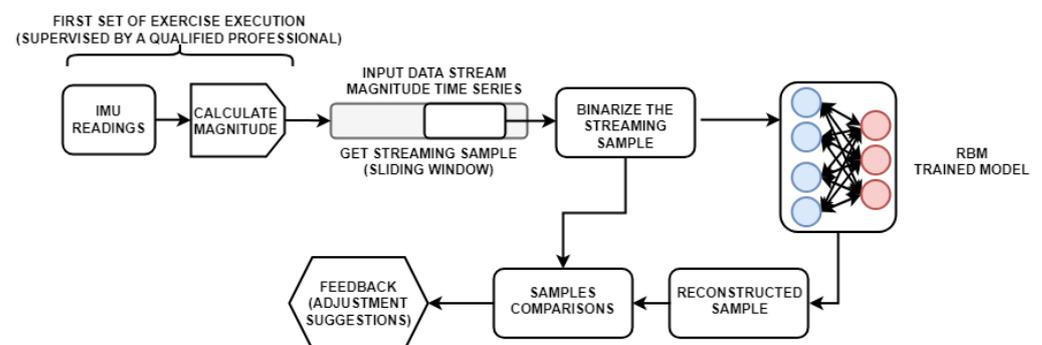


Figure 5. Stream data analysis from proposed method.

The differences between the input binarized sample and the RBM model outputs indicating the points where the evaluated execution *must* be changed to satisfy the correct execution pattern. By comparing the triples associated with each sequential pair from the magnitude series, it is possible to specify if the tendency, existent in the input binary array, is equal to the expected check, if the same units are activated or not. If both 3-tuples have the same units activated, then the series satisfies the model pattern. Otherwise, the output 3-tuple units indicate if the expected values must increase, sustain, or decrease the value.

It is important to stress that the training process demands a high-quality dataset to generate an efficient RBM model. The first set of each physical activity must be executed under the supervision of a qualified professional that guarantees the proper movement execution. This requirement is essential to provide reliable data to use as a reference for correcting the following executions.

5. Evaluation Method

Our performance evaluation consisted of applying a proposed method on a public inertial database, called PHYTMO [20], that contains raw data from NGIMU [40] inertial sensors attached to the subjects’ arms, forearms, thighs, and shins, on both sides (L-left, R-right) as depicted in Figure 6.

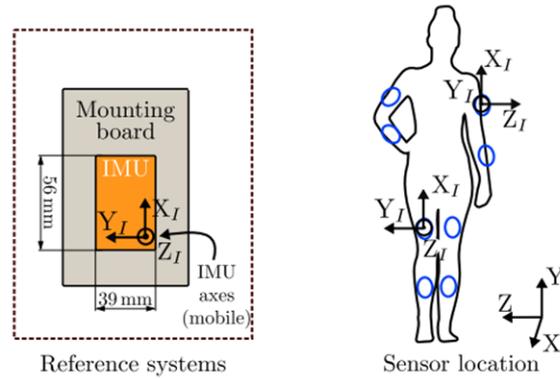


Figure 6. Reference systems indicating the IMU’s axis and the Sensor location indicating the sensor’s placement and its respective axis direction (Adapted from [20]).

This dataset provides data labeled by the groups of the subjects’ ages, exercises, sensor position, and if the movements were performed both, correctly, or incorrectly. This dataset recorded 30 subjects grouped in 5 ranges of age—22 to 26 (A), 30 to 39 (B), 42 to 49 (C), 50 to 55 (D), and 60 to 68 (E)—performing, at least 8×, the following exercises: knee flex–extension (KFE); squats (SQT); hip abduction (HAA); elbow flex–extension (EFE); extension of arms over head (EAH); and squeezing (SQZ).

The raw data from the PHYTMO dataset were converted into magnitude series, following the process explained in the Inertial Sensor Data Section, and these files were used to run the experiments to validate the proposed method. Figure 7 depicts a data sample, after the conversion of raw data into magnitude series, for a single sensor, during the 4 sets of execution where the time series in the Figure 7a,b represent a correct movement execution, and the time series in Figure 7c,d represent incorrect movement execution.

By using the correct execution data to train (Figure 7a) and validate (Figure 7b) the RBM models, and the incorrect execution data (Figure 7c,d) to test them, it is possible to measure the results of applying the recommendations on the incorrect execution samples for each specific body’s segment.

In this paper, this measurement is called the gain metric. It uses the Dynamic Time Warping algorithm to compare a reference series (extracted from the correct movement data) to both the original input series and the output series (generated by the RBM suggestions after evaluating the incorrect movement data). The gain metric is given by $G = DTW(i,r)/DTW(o,r)$, where $DTW(i,r)$ is the distance between i (original input series) and r (reference series), and $DTW(o,r)$ is the distance between the o (output series) and r . This metric expresses how close to the reference series the input series gets after applying the modifications made by the RBM model, in other words, how close to the right movement would be if the users follows the recommendations.

The gain value reflects the following behavior: If the value tends towards 0, the output series will diverge further from the reference series than the original input. When the gain is closer to 1, the output series closely resembles the input series, while values greater than 1 indicate substantial improvements in the output series, reducing the distance from the reference series. In this case, the proposed method aims to maximize the gain value.

The reference series is the sample that provides the smallest average distance value to the other samples from the training data. This means that, for this evaluation, the samples are compared to each other and, then, the one that has the smallest average distance is defined as the reference series.

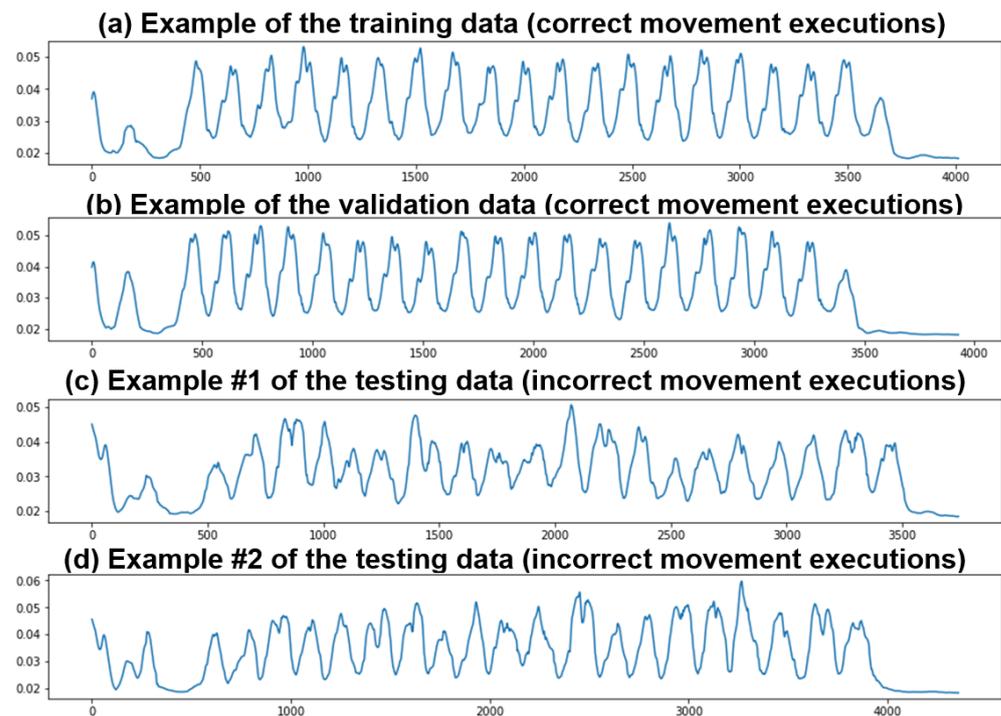


Figure 7. Examples of a training (a), a validation (b) and two testing (c,d) magnitude time series data of an IMU.

The average distance value of the reference series is also used to train a secondary RBM model (RBM+DTW) by excluding the samples that had a distance value greater than the average distance from the reference series. The regular RBM model (RBM) uses unfiltered samples to train the network. Both models are considered in this analysis to highlight the impact of assuming a full set of exercises as a training dataset without filtering the samples. This helps to understand whether the high computational cost of identifying the reference series generates a better model.

Finally, the process described in this Section repeats for all 280 sets (similar to Figure 7). Then, for each set are extracted: (i) a reference sample from the first series of correct exercises; (ii) an RBM model trained from all samples in the first set of correct exercises, and (iii) an RBM+DTW model trained from the samples with distances equal to or smaller than the reference series average distance. The other three series were used to validate the models (second series of correct execution data) and to test the models (both series of incorrect execution data). This evaluation method intends to identify the gain (how similar to the reference series is the output series) after applying the adjustments to movement suggestions, generated by an RBM trained model.

6. Results

This Section is divided into three parts: (i) obtaining the reference series; (ii) validating the models by analyzing a correct movement execution series; and (iii) testing the model on evaluating both incorrect movement execution series.

6.1. Obtaining the Reference Series

Obtaining the reference series has a high computational cost, since we need to compute the distance from all samples to each other. Figure 8 shows the samples extracted from the training series (Figure 7a).

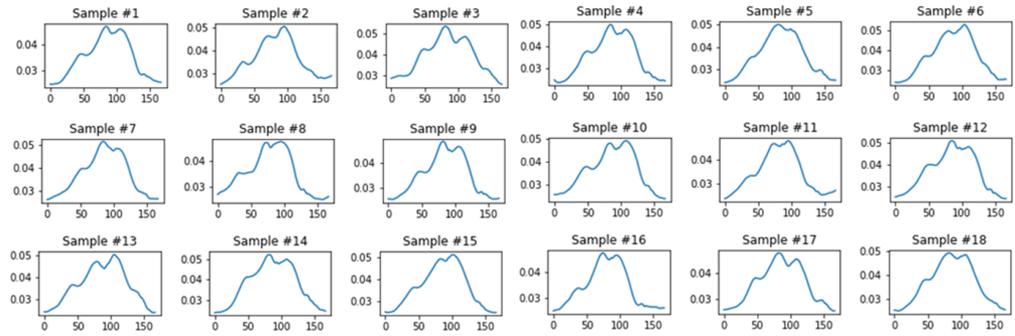


Figure 8. Examples of the samples extracted from the training series.

Each one of the samples depicted in Figure 8 represents a correct movement execution of a physical activity. The reference series must be the sample that generically represents the others, i.e., the most similar sample to the other samples. This similarity is obtained by computing the distance (DTW) of all sample combinations. Table 1 presents the distance of each sample to one another, and the average distance for each one of them. It is possible to see that Sample #5 has the lowest average distance while Sample #3 has the highest average distance value.

It is worth noting that in column #5, some samples (#1, #2, #10, #14, #16, and #17) have distances greater than the average. These samples are not used for training the RBM+DTW model. This filtering creates a more accurate model by training with as highly similar samples as possible.

Table 1. Distance matrix between samples and average distance of a sample to the others (AVG).

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18
#1	0	0.11602	0.212934	0.081604	0.07247	0.14899	0.127803	0.084122	0.040511	0.102565	0.057769	0.093089	0.89771	0.149693	0.113379	0.038214	0.05812	0.07128
#2	0.11602	0	0.124686	0.134002	0.074055	0.3099281	0.069063	0.143369	0.10156	0.122024	0.125735	0.088387	0.09789	0.111101	0.095031	0.123127	0.099294	0.69248
#3	0.212934	0.124686	0	0.186713	0.141121	0.140095	0.080651	0.138278	0.163235	0.17362	0.188406	0.109602	0.179966	0.162766	0.141469	0.164747	0.180205	0.13218
#4	0.081604	0.134002	0.186713	0	0.060254	0.089798	0.117072	0.138855	0.078343	0.079004	0.074407	0.076014	0.073549	0.079045	0.063673	0.076771	0.115931	0.08372
#5	0.07247	0.074055	0.141121	0.060254	0	0.060734	0.061205	0.106907	0.054568	0.074683	0.064012	0.046228	0.064229	0.067814	0.041304	0.065057	0.075678	0.03914
#6	0.14899	0.3099281	0.140095	0.089798	0.060734	0	0.07372	0.199803	0.106341	0.120545	0.115936	0.074533	0.081197	0.05852	0.047283	0.131089	0.152417	0.08918
#7	0.127803	0.069063	0.080651	0.117072	0.061205	0.07372	0	0.12142	0.083682	0.099423	0.107503	0.047224	0.09825	0.086779	0.067423	0.98897	0.091478	0.05404
#8	0.084122	0.143369	0.138278	0.138855	0.106907	0.199803	0.12142	0	0.069673	0.101515	0.069478	0.105539	0.124719	0.207186	0.153632	0.063221	0.76366	0.09591
#9	0.040511	0.10156	0.163235	0.078343	0.054568	0.106341	0.083682	0.069673	0	0.081588	0.065614	0.066284	0.093092	0.114324	0.079407	0.037847	0.057412	0.05052
#10	0.102565	0.122024	0.17362	0.079004	0.074683	0.120545	0.099423	0.101515	0.081588	0	0.084813	0.068209	0.062354	0.115445	0.076069	0.086289	0.093872	0.06801
#11	0.057769	0.125735	0.188406	0.074407	0.064012	0.115936	0.107503	0.069478	0.065614	0.084813	0	0.086766	0.077915	0.111201	0.095041	0.051414	0.07461	0.06498
#12	0.093089	0.088387	0.109602	0.076014	0.046228	0.074533	0.047224	0.105539	0.066284	0.068209	0.086766	0	0.072993	0.075469	0.043741	0.082929	0.085665	0.04905
#13	0.89771	0.09789	0.179966	0.073549	0.064229	0.081197	0.09825	0.124719	0.093092	0.062354	0.077915	0.072993	0	0.082966	0.059379	0.095819	0.118112	0.08591
#14	0.149693	0.111101	0.162766	0.079045	0.067814	0.05852	0.086779	0.207186	0.114324	0.115445	0.111201	0.075469	0.082966	0	0.055455	0.134995	0.151773	0.09816
#15	0.113379	0.095031	0.141469	0.063673	0.041304	0.047283	0.067423	0.153632	0.079407	0.076069	0.095041	0.043741	0.059379	0.055455	0	0.105599	0.115217	0.06746
#16	0.038214	0.123127	0.164747	0.076771	0.065057	0.131089	0.98897	0.063221	0.037847	0.086289	0.051414	0.082929	0.095819	0.134995	0.105599	0	0.06219	0.05825
#17	0.05812	0.099294	0.180205	0.115931	0.075678	0.152417	0.091478	0.76366	0.057412	0.093872	0.07461	0.085665	0.118112	0.151773	0.115217	0.06219	0	0.06125
#18	0.071287	0.69248	0.132188	0.083728	0.039148	0.089181	0.054044	0.095913	0.050529	0.068014	0.064981	0.04905	0.085911	0.098165	0.06746	0.058255	0.061255	0
AVG	0.1370	0.1343	0.1456	0.0894	0.0650	0.0994	0.1320	0.1493	0.0747	0.0894	0.0842	0.0707	0.1314	0.1035	0.0789	0.1315	0.1309	0.1034

6.2. Validation Results

Due to the validation process, the validation series (Figure 7b) were used to evaluate if the models are performing the adjustment recommendation properly. Figure 9 depicts the average distance for each segment.

As expected, the distances from the input series to the reference series have distance values greater than the output series on both models. This means that the suggestions generated by both the RBM and RBM+DTW models suggested changes that make the input more similar to the reference series.

Figure 10 is a sample that compares the input series (Figure 10a), the RBM output series (Figure 10b), and the RBM+DTW output series (Figure 10c) to the reference series (blue line in all of them). It is worth noting that the RBM model generates a gain of 48.15% and the RBM+DTW model generates an output series that has a gain of 32.38%, compared to the input series, which means that both output series are 48.15% and 32.38% more similar to the reference series than the original input.

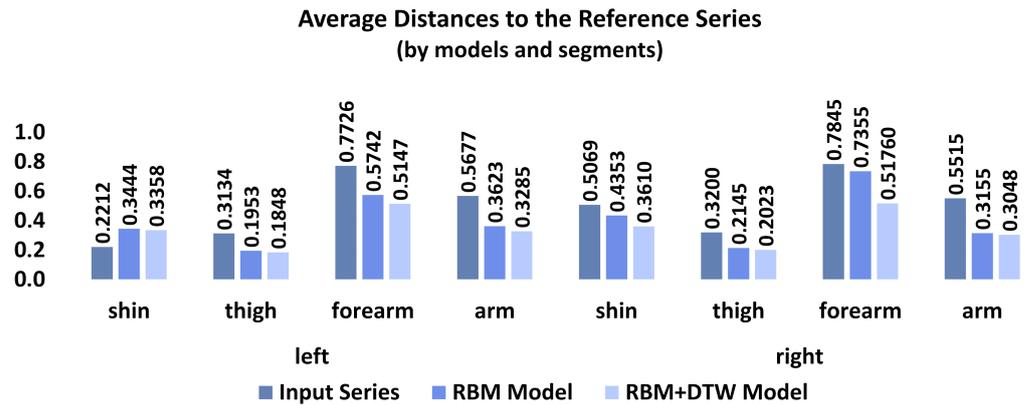


Figure 9. Validation results: average distance to the reference series by body segment.

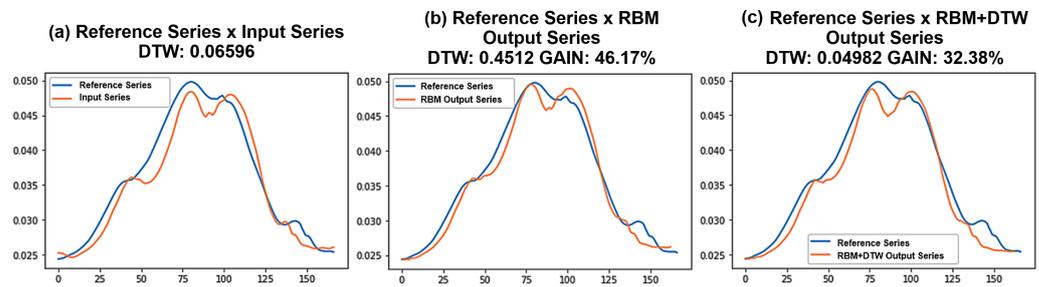


Figure 10. Validation, good results: sample of the comparisons between the reference series and: (a) input series; (b) RBM output series; and (c) RBM+DTW output series.

An interesting result could be observed in Figure 10, in which the suggestions for the sensor placed on the left shin are further away than the original input data. As the validation process uses correct execution data, this result could be interpreted as an input series that already satisfies the prior distribution learned from the RBM. Figure 11 depicts that behavior, where the output series is 13.73% and 12.94%, far from the original input, for the RBM and the RBM+DTW models, respectively.

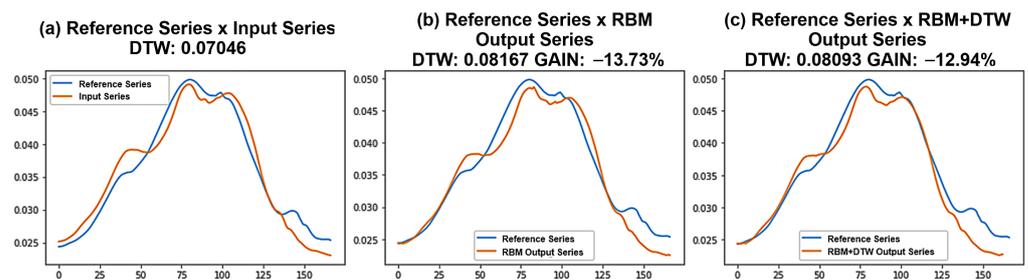


Figure 11. Validation, bad results: Sample of the comparisons between the reference series and: (a) input series; (b) RBM output series; and (c) RBM+DTW output series.

Despite the observed behavior, the adjustment suggestions did not generate outputs series that change the series to a variation that could be considered an incorrect movement, preserving the patterns of the reference series.

Finally, Figure 12 shows the average gain by the body’s segment when applying the model’s suggestions. The results show that RBM+DTW outperformed the RBM model, making it a more accurate model, providing efficient suggestions.

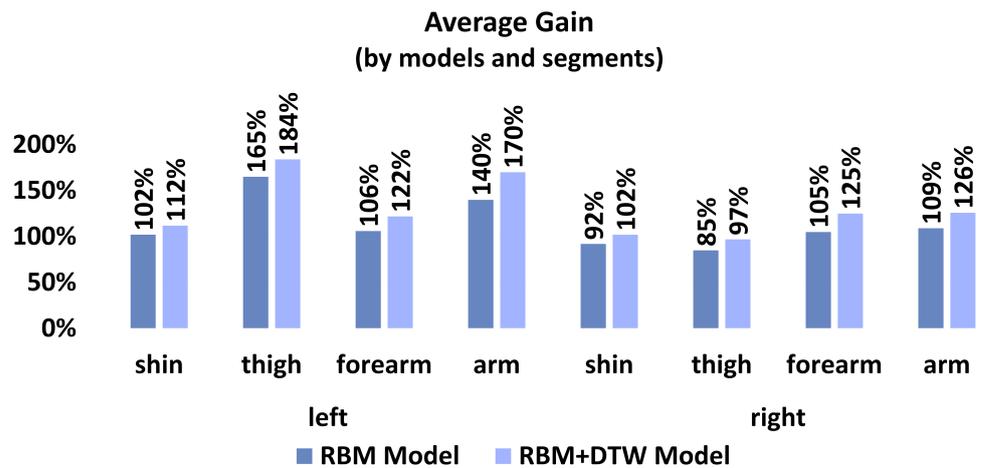


Figure 12. Validation: average gain of each model by segment.

The previous conclusion does not disqualify the RBM model. Despite the inferior performance compared to the RBM+DTW models, the RBM model still generates an average gain of 85% in relation to the input series for the thigh sensor, which means the RBM output series are 1.7× more similar to the reference series than the input series. The RBM+DTW models generate outputs up to 184% better than the input (left thigh segment).

The validation process satisfied our purpose by presenting the expected results when analyzing correct movement data.

6.3. Testing Results

Differently from the validation process, the testing experiments used incorrect execution data to evaluate the models. The expected outcome was to achieve better results than the validation process, as the models would adjust the incorrect executions, leading to even greater gains.

In Figure 13, it is possible to observe the tendency, previously shown in the validation process, that the output series were closer to the reference series than the original input, even on the left shin sensor.

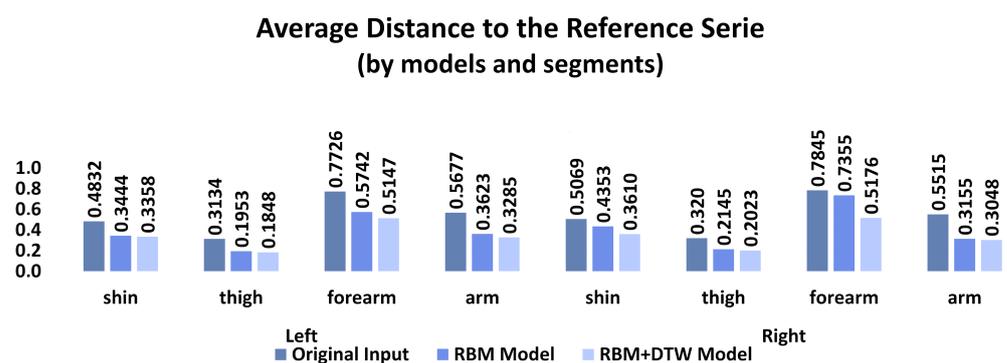


Figure 13. Average distance of each model by segment.

The average gain for each model, presented in Figure 14, starts from 106% for the RBM model, up to 232% for the RBM+DTW model. Those results express the efficiency of following the suggestions made by the models. This means that both models generate output series at least 2× better than the original input.

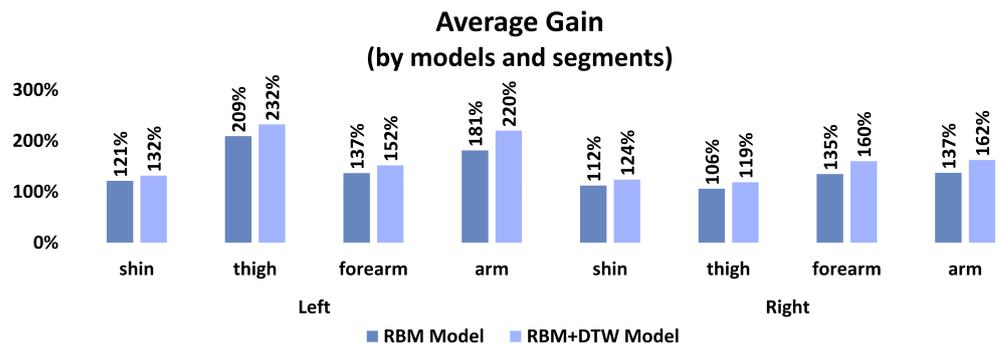


Figure 14. Testing: average gain of each model by segment.

An example of how both models, RBM and RBM+DTW, correct the input series could be observed in Figure 15. The input series contains a decreasing tendency at the beginning and an increasing tendency at the end (Figure 15a). These tendencies do not exist in the reference series (see the reference series in the Figures 8, 10 and 11).

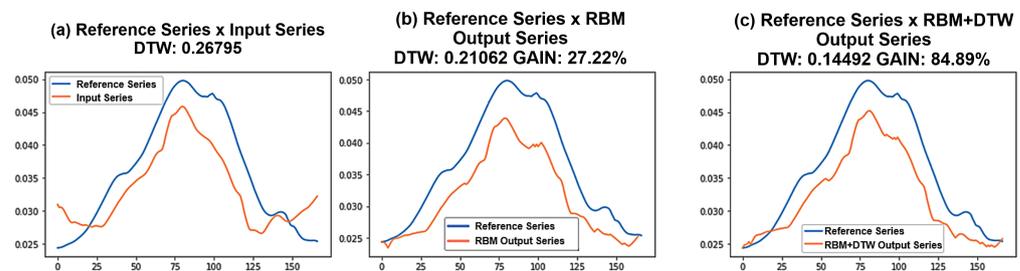


Figure 15. Testing results: example of the comparisons between the reference series and: (a) input series; (b) RBM output series; and (c) RBM+DTW output series.

The RBM model suggests adjustment on the extremities of the input series, but it creates a bigger gap between the series (Figure 15b). The RBM+DTW models also change the input extremity but obtain better results on preserving a smaller gap than the reference series and the RBM+DTW output series (Figure 15c). This example shows that the RBM models generate an output series 27.22% better (more similar to the reference series) than the input series, against the 84.89% from the RBM+DTW output series.

In Figure 16, it is possible to compare the gain of both models by exercise and body segment. As explained before, the RBM+DTW outperforms the RBM with a higher gain value, sustaining the previous analysis.

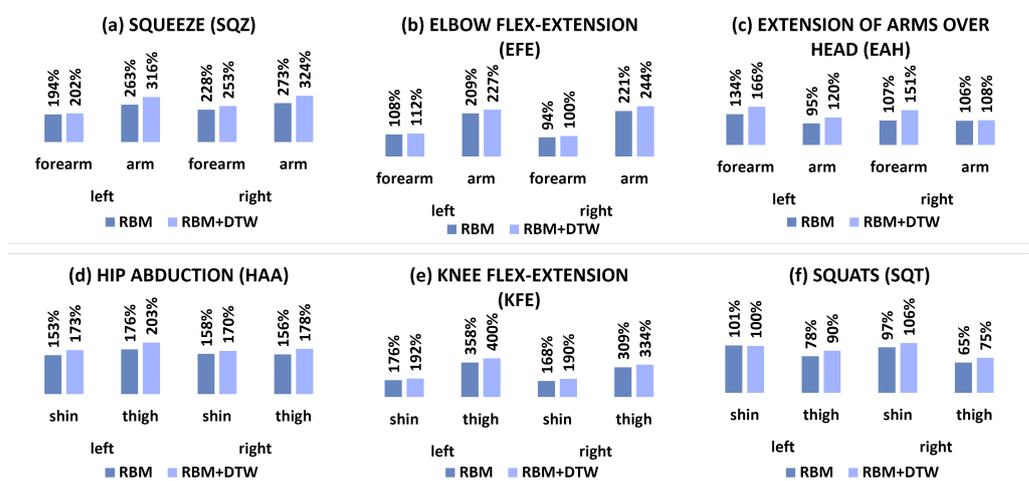


Figure 16. Gain by model on each exercise for each body segment.

An interesting result could be observed on elbow flex–extension (Figure 16b), where the gains for the arms are greater than the forearm, in both sizes. This could be explained by the difference between the movements on each segment. During the EFE, the arms move less than the forearms, which means that any suggestion (modification) on the arms has a huge impact on the gain. This is different from a long movement which accepts some small variation based on the training process. This same behavior could be observed on knee flex–extension (Figure 16e), where the shins move longer than the thighs along the exercise execution.

The RBM model during the squats (Figure 16f) exercise has the smallest average gain of all exercises, while the RBM+DTW produced the highest average gain during the knee flex–extension (Figure 16e) exercise. Figure 14 highlights the gain on use of these models during the exercises squeeze (Figure 16a) and knee flex–extension (Figure 16b), where both produced suggestions with gains higher than 200%, while the squat (Figure 16f) produced gains smaller than 90% but greater than 50%. This means that, even in the worst case of this approach (Figure 16f), the use of the proposed method presents an output series at least 65% better than the input series.

6.4. Discussion

Through the results presented in Section 6, it was possible to address the motivating questions of this article, where: (i) By generating a machine learning model using a Restricted Boltzmann Machine, from a binarized time series, that represents the inertial movements of each body segment, it is possible to indicate which points, in the input series, do not satisfy the data patterns from the training process and corrects them in such a way that they approach the expected pattern by the neural network. (ii) By applying the proposed model to a database containing inertial data from subjects performing various physical exercises correctly and incorrectly, it was possible to train a model with the correct data and recommend adjustments in the time series of the incorrect data, which allowed an increase of up to $3.6\times$ in the similarity between the input series and the output series generated by the machine learning model. (iii) The method suggests that the creation of highly specific models that can identify the user's movement patterns at the beginning of each physical exercise. This feature ensures that the model which analyzes the subsequent repetitions is as up-to-date as possible. This method is flexible enough to adapt to any repetitive physical activities.

By analyzing the results, it was possible to identify that despite the use of DTW to find a reference series between the training sample has a high computational cost, this mechanism presented relevant advantages when used as a filter to select the samples for training the models. As presented in Figure 14, the RBM+DTW model outperformed the gains of the RBM for each segment.

Another interesting insight from the experimental results is that, eventually, the input series already satisfies the training pattern; in this case, both the RBM and RBM+DTW models generate output series which are slightly different from the input series, which may increase the distance to the reference series. In this case, it is highly recommended to ignore the suggestions. In other words, if the input series already satisfies the pattern distribution from the training data, it is not necessary to submit this input series to the models.

A third observation from the experiment relies on the use of the magnitude metric, which demonstrates itself as a powerful mechanism to reduce the model dimensionality. Assuming the IMU readings occur in a specific frequency (i.e., 10 Hz or 10 readings per second), it is possible to estimate the movement duration and its velocity by observing the variation between the readings where higher differences means faster movements and lower differences means slower movements.

Finally, a fourth insight from these experiments is that by assuming the number of elements in the reference series as the default sliding window size, it is possible to infer an input series that does not complete the movement inside this window as a wrong movement execution; this assumption relies on the fact that if the movement does not finish

along the window size, then it was performed too slowly. On the other hand, multiple executions inside the same sliding window means fast executions, which also does not satisfy the training pattern.

7. Conclusions

The proposed method presents an alternative to evaluate repetitive physical activities and provide suggestions to improve the user's performance by creating highly specialized models. The main goal is to provide a novel method for movement recognition and evaluation.

By employing a metric known as 'magnitude', which represents the absolute values of readings from the inertial sensors, we were able to utilize a Restricted Boltzmann Machine for assessing movement tendencies and providing suggestions for new inputs that deviated from the patterns learned in the training dataset. Differently from the usual classification of neural networks, this approach aims to adjust the input data to an output that satisfies the RBM model. The differences between the input and the output data indicate the points where the movement must change to be performed correctly. The proposed method mitigates the sensor displacement problem by retraining the model with the updated sensor position and assuming absolute values regardless of the axis position on the body's segment.

Although the results have shown that using unfiltered samples to generate the RBM model leads to less accurate suggestions compared to RBM+DTW, this approach still produced output series that, on average, were $1.7\times$ closer (or similar) to the reference series than the input series. In contrast, the RBM+DTW model generated output series that were up to $3.68\times$ superior.

It is important to highlight the computational cost for RBM+DTW, to identify the reference series, which is a computationally expensive operation. Such a method could hardly be used on embedded devices like smartwatches due to resource constraints. On the other hand, the unfiltered model has the potential to do so. The possible applications of this method could span across several areas, such as gym activities monitoring, physiotherapy, robotic arm calibration, and remote activity monitoring. The studies presented in [41] provide a preliminary approach to adapt the proposed model for integration into wearable devices.

As a major limitation, the application of this methodology in a real scenario demands a qualified professional responsible for guaranteeing the quality of the training data; this means that a user must understand and comprehend the physical activity execution and try to repeat them as best as possible according to the professional orientation during the training set. Also, it is not recommended to reuse a trained model for monitoring executions in different training sessions, due to the risk of the sensor being attached to the wrong segment or the user's limitation/rhythm change between the session, which may generate incorrect feedback. Another challenge of this approach is to define when an input series (from streaming data) must be analyzed by the RBM model. It is not suitable nor practical to submit a series to the model at each new sensor's reading. The criterion to evaluate the stream buffer used in this paper is the max value in the sliding window; when it reaches the same index as the max value on the reference series, then the input series is binarized and submitted to the RBM model for evaluation. Finally, the magnitude metric suppresses the capacity to identify the movement orientation and angles. However, the magnitude eliminates the requirement to position the sensors in a specific orientation.

To the best of our present knowledge, the literature does not have suitable solutions to address the problem of providing qualitative feedback for helping users to improve their physical movements. Pushing the limits of the state of the art in human activity recognition, there are some relevant studies such as [42,43] that use deep neural networks to recognize multiple human daily activities. Other works such as [44] use hybrid techniques to combine multiple data sources to recognize human activity. Finally, some studies such as [45] explore deep learning techniques to identify human walking activities. Nonetheless,

the method presented in this paper is distinguished because most existing approaches in the literature concentrate solely on recognizing particular actions or movements, without offering feedback to help users correct a wrong movement and make it right. Furthermore, the proposed method offers several advantages, such as: (i) enhanced technique, in the sense that the system helps users perform exercises correctly, ensuring that the technique is appropriate; (ii) real-time feedback, where users receive immediate feedback while performing exercises, helping them correct any error as they occur, rather than afterward; (iii) by providing guidance on improper movements, the system can help prevent injuries caused by poorly executed exercises; and (iv) as the system can be used by a wide range of people in different locations, making correct exercises more accessible.

Finally, further studies could combine multiple suggestions from each segment sensor as a single one. By combining these multiple outputs into unique feedback, it is possible to evaluate a complex system using smaller evaluations as a divide-and-conquer approach. Also, as the next step, a deeper comparison of the proposed method to the state-of-the-art in data reconstruction focusing on time series.

Author Contributions: Conceptualization, investigation, methodology and validation, M.A. and R.B.; Writing (original, review and editing), M.A., R.B., E.S. and H.O.; Supervision, R.B.; Funding Acquisition, E.S. and H.O. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES-PROEX)—Finance Code 001. This work was partially supported by Amazonas State Research Support Foundation—FAPEAM—through the POSGRAD 2022-2023 project. This research, according to Article 48 of Decree n° 6.008/2006, was partially funded by Samsung Electronics of Amazonia Ltda, under the terms of Federal Law n° 8.387/1991, through agreement n° 003/2019, signed with ICOMP/UFAM.

Data Availability Statement: All the experiment results data in this manuscript are available on [46]. The original dataset (PHYTMO [20]) used for validating and testing the model is available on <https://zenodo.org/record/6319979> (accessed on 16 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RBM	Restricted Boltzmann Machines
DTW	Dynamic time warping
SVM	Support vector machine
KFE	Knee flex–extension
SQT	Squats
HAA	Hip abduction
EFE	Elbow flex–extension
EAH	Extension of arms over head
SQZ	Squeezing

References

1. Lima, W.S.; Bragança, H.L.; Souto, E.J. NOHAR—NOvelty discrete data stream for Human Activity Recognition based on smartphones with inertial sensors. *Expert Syst. Appl.* **2021**, *166*, 114093. [CrossRef]
2. Amir, N.I.M.; Dziauddin, R.A.; Mohamed, N.; Ismail, N.S.N.; Zulkifli, N.S.A.; Din, N.M. Real-time Threshold-Based Fall Detection System Using Wearable IoT. In Proceedings of the 2022 4th International Conference on Smart Sensors and Application (ICSSA), Kuala Lumpur, Malaysia, 26–28 July 2022; pp. 173–178. [CrossRef]
3. Liu, C.; Han, L.; Chang, S.; Wang, J. An Accelerometer-Based Wearable Multi-Node Motion Detection System of Freezing of Gait in Parkinson’s Disease. In Proceedings of the 2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Ottawa, ON, Canada, 16–19 May; IEEE Press: New York, NY, USA, 2022; pp. 1–5. [CrossRef]
4. Fuller, D.; Ferber, R.; Stanley, K. Why machine learning (ML) has failed physical activity research and how we can improve. *BMJ Open Sport Exerc. Med.* **2022**, *8*, e001259. [CrossRef]

5. Pavlov, K.; Perchik, A.; Tsepulin, V.; Megre, G.; Nikolaev, E.; Volkova, E.; Nigmatulin, G.; Park, J.; Chang, N.; Lee, W.; et al. Sweat Loss Estimation Algorithm for Smartwatches. *IEEE Access* **2023**, *11*, 23926–23934. [\[CrossRef\]](#)
6. Montull, L.; Slapšinskaitė-Dackevičienė, A.; Kiely, J.; Hristovski, R.; Balagué, N. Integrative Proposals of Sports Monitoring: Subjective Outperforms Objective Monitoring. *Sport. Med.-Open* **2022**, *8*, 1–10. [\[CrossRef\]](#)
7. Chen, Y.; Pei, M.; Nie, Z. Recognizing Activities from Egocentric Images with Appearance and Motion Features. In Proceedings of the 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP), Gold Coast, Australia, 25–28 October 2021; pp. 1–6. [\[CrossRef\]](#)
8. Gu, Y.; Song, Y.; Goncharenko, I.; Kamijo, S. Driver Hand Activity Recognition using NIR Camera and Deep Neural Network. In Proceedings of the 2022 IEEE 4th Global Conference on Life Sciences and Technologies (LifeTech), Osaka, Japan, 7–9 March 2022; pp. 299–300. [\[CrossRef\]](#)
9. Zhang, T.; Su, Z.; Cheng, J.; Xue, F.; Liu, S. Machine vision-based testing action recognition method for robotic testing of mobile application. *Int. J. Distrib. Sens. Netw.* **2022**, *18*, 155013292211153. [\[CrossRef\]](#)
10. Zhu, M.; Guan, X.; Wang, Z.; Qian, B.; Jiang, C. sEMG-Based Knee Joint Angle Prediction Using Independent Component Analysis & CNN-LSTM. In Proceedings of the 2022 6th International Conference on Measurement Instrumentation and Electronics (ICMIE), Hangzhou, China, 17–19 November 2022; pp. 13–18. [\[CrossRef\]](#)
11. Khant, M.; Lee, D.T.; Gouwanda, D.; Gopalai, A.A.; Lim, K.H.; Foong, C.C. A Neural Network Approach to Estimate Lower Extremity Muscle Activity during Walking. In Proceedings of the 2022 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), Kuala Lumpur, Malaysia, 7–9 December 2022; pp. 106–111. [\[CrossRef\]](#)
12. Wang, X.; Hao, M.; Chou, C.h.; Zhang, X.; Pan, Y.; Sun, B.; Bai, M.; Dai, C.; Lan, N. The Effects of Deep Brain Stimulation on Motor Unit Activities in Parkinson’s Disease based on High-Density Surface EMG Analysis. In Proceedings of the 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Glasgow, UK, 11–15 July 2022; pp. 682–685. [\[CrossRef\]](#)
13. Botros, F.S.; Phinyomark, A.; Scheme, E.J. Electromyography-Based Gesture Recognition: Is It Time to Change Focus From the Forearm to the Wrist? *IEEE Trans. Ind. Inform.* **2022**, *18*, 174–184. [\[CrossRef\]](#)
14. Lin, Q.; Wu, Y.; Liu, J.; Hu, W.; Hassan, M. Demo Abstract: Human Activity Detection with Loose-Fitting Smart Jacket. In Proceedings of the 2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Sydney, Australia, 21–24 April 2020; pp. 361–362. [\[CrossRef\]](#)
15. Centracchio, J. A New Piezoelectric Sensor for Forcemycography Application. In Proceedings of the 2022 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 17–18 November 2022; pp. 1–4. [\[CrossRef\]](#)
16. Kim, J.; Kwak, Y.H.; Kim, W.; Park, K.; Pak, J.J.; Kim, K. Flexible force sensor based input device for gesture recognition applicable to augmented and virtual realities. In Proceedings of the 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, South Korea, 28 June–1 July 2017; pp. 271–273. [\[CrossRef\]](#)
17. Octavian, C.A.; Mihaela, H.; Catalin, I.J. Gesture Recognition using PYTHON. In Proceedings of the 2021 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Brasov, Romania, 18–21 May 2021; pp. 139–144. [\[CrossRef\]](#)
18. Patil, A.K.; Balasubramanyam, A.; Ryu, J.; Chakravarthi, B.; Chai, Y.H. An Open-Source Platform for Human Pose Estimation and Tracking Using a Heterogeneous Multi-Sensor System. *Sensors* **2021**, *21*, 2340. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Guinea, A.S.; Sarabchian, M.; Mühlhäuser, M. Image-based Activity Recognition from IMU Data. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Kassel, Germany, 22–26 March 2021; pp. 14–19. [\[CrossRef\]](#)
20. Villa, S.; Jiménez Martín, A.; García, J. A database of physical therapy exercises with variability of execution collected by wearable sensors. *Sci. Data* **2022**, *9*, 266. [\[CrossRef\]](#)
21. Vintsyuk, T.K. Speech discrimination by dynamic programming. *Cybernetics* **1968**, *4*, 52–57. [\[CrossRef\]](#)
22. Sakoe, H.; Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 43–49. [\[CrossRef\]](#)
23. Rumelhart, D.E.; McClelland, J.L. Information Processing in Dynamical Systems: Foundations of Harmony Theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*; MIT Press: Cambridge, MA, USA, 1987; pp. 194–281.
24. Li, W.; Han, M.; Wang, J. Recurrent Restricted Boltzmann Machine for Chaotic Time-series Prediction. In Proceedings of the 2020 12th International Conference on Advanced Computational Intelligence (ICACI), Yunnan, China, 14–16 March 2020; pp. 439–445. [\[CrossRef\]](#)
25. Conforti, I.; Mileti, I.; Del Prete, Z.; Palermo, E. Measuring Biomechanical Risk in Lifting Load Tasks Through Wearable System and Machine-Learning Approach. *Sensors* **2020**, *20*, 1557. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Liao, Y.; Vakanski, A.; Xian, M. A Deep Learning Framework for Assessing Physical Rehabilitation Exercises. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2020**, *28*, 468–477. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Huamin, T.; Qiuqun, D.; Shanzhu, X. Reconstruction of time series with missing value using 2D representation-based denoising autoencoder. *J. Syst. Eng. Electron.* **2020**, *31*, 1087–1096. [\[CrossRef\]](#)
28. De Villa, S.G.; Parra, A.M.; Martín, A.J.; Domínguez, J.J.G.; Casillas-Perez, D. ML algorithms for the assessment of prescribed physical exercises. In Proceedings of the 2021 IEEE International Symposium on Medical Measurements and Applications (MeMeA), Lausanne, Switzerland, 23–25 June 2021; pp. 1–6. [\[CrossRef\]](#)

29. Chen, K.Y.; Shin, J.; Hasan, M.A.M.; Liaw, J.J.; Yuichi, O.; Tomioka, Y. Fitness Movement Types and Completeness Detection Using a Transfer-Learning-Based Deep Neural Network. *Sensors* **2022**, *22*, 5700. [[CrossRef](#)] [[PubMed](#)]
30. Mekruksavanich, S.; Jitpattanakul, A. Exercise Activity Recognition with Surface Electromyography Sensor using Machine Learning Approach. In Proceedings of the 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), Pattaya, Thailand, 11–14 March 2020; pp. 75–78. [[CrossRef](#)]
31. Saidani, S.; Haddad, R.; Bouallegue, R. A prototype design of a smart shoe insole system for real-time monitoring of patients. In Proceedings of the 2020 6th IEEE Congress on Information Science and Technology (CiSt), Agadir, Morocco, 5–12 June 2020; pp. 116–121. [[CrossRef](#)]
32. Fortes Rey, V.; Garewal, K.K.; Lukowicz, P. Translating Videos into Synthetic Training Data for Wearable Sensor-Based Activity Recognition Systems Using Residual Deep Convolutional Networks. *Appl. Sci.* **2021**, *11*, 3094. [[CrossRef](#)]
33. Shoaib, M.; Bosch, S.; Incel, O.; Scholten, H.; Havinga, P. Fusion of Smartphone Motion Sensors for Physical Activity Recognition. *Sensors* **2014**, *14*, 10146–10176. [[CrossRef](#)] [[PubMed](#)]
34. Kwolek, B.; Kepski, M. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Comput. Methods Programs Biomed.* **2014**, *117*, 489–501. [[CrossRef](#)]
35. Chu, Y.; Zhao, X.; Zou, Y.; Xu, P.; Han, J.; Zhao, Y. A Decoding Scheme for Incomplete Motor Imagery EEG With Deep Belief Network. *Front. Neurosci.* **2018**, *12*, 680. [[CrossRef](#)]
36. Al-Salman, O.; Mustafina, J.; Shahoodh, G. A Systematic Review of Artificial Neural Networks in Medical Science and Applications. In Proceedings of the 2020 13th International Conference on Developments in eSystems Engineering (DeSE), Virtual Conference, 14–17 December 2020; pp. 279–282. [[CrossRef](#)]
37. K, R.C.; Dhananjaya, G.; Ds, L.; Krishnamurthy, P.L.; Shreya Pawar, A. Recommendation System Using Deep Learning. In Proceedings of the 2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Surathkal, India, 1–3 December 2022; Volume 7, pp. 99–103. [[CrossRef](#)]
38. Zohra, M.F.; Artaa, R.; Asma, Z.; Benmerzoug, D. Predicted Model based on Boltzmann Restricted Machine for Web Services Recommendation. In Proceedings of the 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS), Bouaghi, Algeria, 12–13 October 2022; pp. 1–7. [[CrossRef](#)]
39. Tavenard, Romain. An introduction to Dynamic Time Warping. 2021. Available online: <https://rtavenar.github.io/blog/dtw.html> (accessed on 21 September 2023).
40. Technologies, X.I. NGIMU—Wearable Sensor. Available online: <https://x-io.co.uk/ngimu/> (accessed on 15 May 2023).
41. Alencar, M.; Barreto, R.; Oliveira, H.; Souto, E. Embedded Restricted Boltzmann Machine Approach for Adjustments of Repetitive Physical Activities Using IMU Data. *IEEE Embed. Syst. Lett.* **2023**, *2023*, 1. [[CrossRef](#)]
42. Bijalwan, V.; Semwal, V. Wearable sensor based Pattern Mining for Human Activity Recognition : Deep Learning Approach. *Ind. Robot.* **2021**, *48*, 187. [[CrossRef](#)]
43. Jain, R.; Semwal, V.; Kaushik, P. Deep ensemble learning approach for lower extremity activities recognition using wearable sensors. *Expert Syst.* **2022**, *38*, 12743. [[CrossRef](#)]
44. Bijalwan, V.; Semwal, V.; Singh, G.; Mandal, T. HDL-PSR: Modelling Spatio-Temporal Features Using Hybrid Deep Learning Approach for Post-Stroke Rehabilitation. *Neural Process. Lett.* **2022**, *54*, 279–298. [[CrossRef](#)]
45. Semwal, V.; Gaud, N.; Lalwani, P.; Bijalwan, V.; Alok, A. Pattern identification of different human joints for different human walking styles using inertial measurement unit (IMU) sensor. *Artif. Intell. Rev.* **2022**, *55*, 1149–1169. [[CrossRef](#)]
46. Alencar, M. Experiments Results. 2023. Available online: https://github.com/macalencar/PhysicalExercisesRBM_Results (accessed on 15 May 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.