A Deep Learning Model for Exercise-Based Rehabilitation Using Multi-channel Time-Series Data from a Single Wearable Sensor

Ghanashyama Prabhu[1,2,4(✉)], Noel E. O'Connor[1,2], and Kieran Moran[1,3]

[1] Insight SFI Research Centre for Data Analytics, Dublin City University, Dublin, Ireland
ghanashyama.prabhu2@mail.dcu.ie, gs.prabhu@manipal.edu
[2] School of Electronic Engineering, Dublin City University, Dublin, Ireland
[3] School of Health and Human Performance, Dublin City University, Dublin, Ireland
[4] Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India
https://www.insight-centre.org/

Abstract. The ability to accurately and automatically recognize and count the repetitions of exercises using a single sensor is essential for technology-assisted exercise-based rehabilitation. In this paper, we present a single deep learning architecture to undertake both of these tasks based on multi-channel time-series data. The models are constructed and tested using the INSIGHT-LME [1] exercise dataset which consists of ten local muscular endurance (LME) exercises. For exercise recognition, we achieved an overall F1-score measure of 96% and for repetition counting, we were correct within an error of ±1 repetitions in 88% of the observed exercise sets. To the best of our knowledge, our approach of using the same deep learning model for both tasks using raw time-series sensor data information is novel.

Keywords: INSIGHT-LME dataset · CNN · Wearable sensor · Exercise-based rehabilitation · Multi-channel time-series

1 Introduction

Community-based or home-based exercising are approaches commonly adopted for rehabilitation. Exercise-based rehabilitation often needs to be long-term. Unfortunately, for a variety of reasons (including travel distances, organized classes not being schedule-friendly and some people not wanting to exercise in front of others) adherence to organised programmes tend to be very low [2,3]. Alternatively, if people could exercise anywhere convenient to them, at any time,

Supported by Insight SFI Research Centre for Data Analytics.

J. Ye et al. (Eds.): MobiHealth 2020, LNICST 362, pp. 104–115, 2021.
https://doi.org/10.1007/978-3-030-70569-5_7

it may act to motivate the uptake and adherence to exercise-based rehabilitation. Such an approach would be facilitated if information on the type and amount of exercise was automatically detected for real-time and summary feedback, which has been shown to be a motivating factor rehabilitating patients. Technology advances in wearable sensors have resulted in cost-effective devices capable of recording human movements effectively [4,5]. Human activity recognition (HAR) is an increasingly important research topic where human movements and associated activities are studied using advanced artificial intelligence algorithms, e.g. machine learning and deep learning models, applied to sensor data from wearables. In recent years, the use of a single wearable sensor has gained prominence in different areas of HAR such as: day-to-day activity(e.g. jogging, running, walking, drinking, sitting) [6–9], gym activity [10] and exercise [11–14] recognition and in repetition counting [11,15,16]. Studies have shown that elderly rehabilitation patients (about 68%) have indicated their interest in using a single sensor (inertial measurement unit) within exercise-based rehabilitation [2].

The increased interest in using deep learning models in the field of HAR and especially exercise [1,11,17] has resulted in various models being used for exercise recognition and repetition counting. However, it appears that no studies have used a single deep CNN model architecture using multi-channel time-series data for exercise recognition and repetition counting. Using a single model architecture for both tasks simplifies implementation and training. This is an important consideration if the AI-based technique were ultimately to be implemented as an embedded function of the wearable sensing platform. As such, this study aims to demonstrate how a single CNN model architecture can be used for automatic exercise recognition and repetition counting using multi-channel time-series data obtained from a single inertial measurement unit.

2 Proposed Framework

Figure 1 represents the end-to-end pipeline framework used for the exercise recognition and repetition counting. This framework consists of a data processing unit, two CNN models and an output processing component. The data processing unit processes the INSIGHT-LME dataset [1] into 6D time-series arrays. Two CNN models were constructed using a single architecture for both the exercise recognition and the repetition counting tasks. The output processor consists of two fully connected layers, the first one is used at the output of the CNN model for exercise recognition and the second one is used at the output of the CNN model for repetition counting.

3 Methodology

3.1 Data Set

We have used the INSIGHT-LME dataset, a data set recently made publicly available (https://bit.ly/30UCsmR), consisting of eleven classes of movements with first ten classes corresponding to ten LME exercises commonly used in

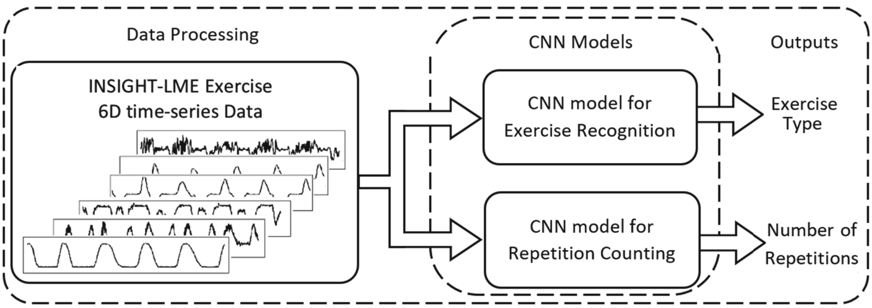
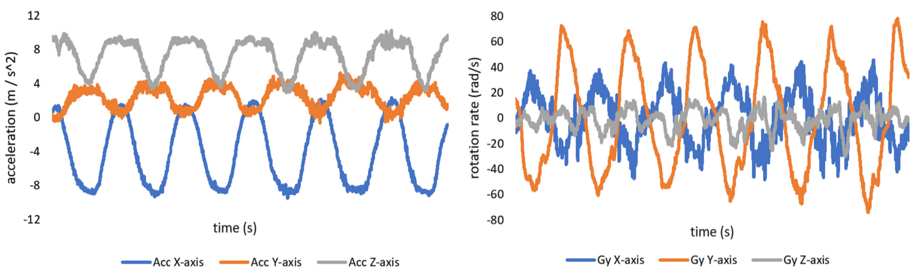


Fig. 1. End-to-End pipeline for exercise recognition and repetition counting.

exercise-based cardiovascular disease (CVD) rehabilitation and the eleventh class corresponding to movements commonly observed between exercises. The ten LME exercises consists of six upper-body LMEs (*Bicep Curls* (BC), *Frontal Raise* (FR), *Lateral Raise* (LR), *Triceps Extension Right arm* (TER), *Pec Dec* (PD) and *Trunk Twist* (TT)), and four lower-body LMEs (*Squats* (SQ), *Lunges* (L), *Leg Lateral Raise* (LLR) and *Standing Bicycle Crunches* (SBC)). The dataset consists of raw time-series data from a 3D accelerometer and a 3D gyroscope using a single inertial measurement unit (IMU) mounted on the right-wrist and was collected from 76 healthy and able bodied participants. The IMUa used in the dataset was Shimmer3 IMUs which were light-weight wearable sensor units from Shimmer[1]. Each IMU used in the data collection process was calibrated using Shimmer's 9DoF calibration application[2] and a sampling rate 512 Hz was used. Exercise data were collected in two sets from the participants under constrained and unconstrained environments. 6D time-series data (3D accelerometer and 3D gyroscope) were further used in the data processing. As an illustrative example, Fig. 2 represents 25 s segmented time-series sensor signal plots of 3D accelerometer and 3D gyroscope for the *Frontal Raise* exercise.



(a) 3D Accelerometer plot (b) 3D Gyroscope plot

Fig. 2. 25 s segmented plots of Frontal Raise exercise

[1] http://www.shimmersensing.com/products/shimmer3.
[2] https://www.shimmersensing.com/products/shimmer-9dof-calibration.

3.2 Data Processing

Data processing was performed on the INSIGHT-LME dataset to have 6D time-series array information with two target labels. The new 6D time-series information was generated from data segmentation process using a sliding window method. A window-length of 4 s and an overlap of 0.5 s was used in data segmentation process. From every 4 s segment of exercise data, a 6D time-series data array was formed and was computed for all exercise data. The processed data, from 76 participants, was divided into three subsets. A training set was formed with data from 46 participants. Additionally, from the remaining participants a test set and a validation set were formed with data from 15 participants each.

The two class labels were generated for the new 6D time-series information. First target labels were used for the exercise recognition task and the second target labels were used in the repetition counting task. The first target labels were for the exercise recognition task and were the eleven class label information of the exercise movements. However, for the repetition counting task, a new binary class label was added on each 4 s segmented array data using a 50% grid method. Ground truth with the newer binary class information was generated using dominant signal information for each exercise [1,16,18]. If the dominant signal peak lay at the left half of the grid then a label information "Peak" (or "1") was added, otherwise "No Peak" (or "0") label information was added.

3.3 A Deep CNN Architecture for Recognition and Repetition Counting

HAR recognition, especially in the field of exercise recognition and repetition counting, few recent studies [1,11,17] have used different deep CNN structures. A single CNN architecture was used by [11] which uses one model for exercise recognition but uses ten different models for repetition counting. However, in our previous study [1] we have successfully demonstrated building two models using the state of the art AlexNET architecture, one for all the exercise recognition and the other for repetition counting from all the exercises in contrast to Soro et al. [11]. However, it appears that no studies have used a single deep CNN model architecture using multi-channel time-series data for exercise recognition and repetition counting.

We designed and built deep CNN models from scratch using the same base structure (Fig. 3), one for the exercise recognition and other for the repetition counting. The architecture consists of seven 2D convolutional layers (*ConvLayer*) in addition to an input layer, two fully connected layers and a dropout layer. The number of filters used in seven convolution layers were 16, 16, 32, 32, 64, 64 and 96 respectively. The selection of the number of convolutional layers and the number of filters in each layer of the CNN_Model2 architecture were arrived after the initial few trials with different configurations. Output of each *ConvLayer* was batch normalized [19] and rectified linear units (ReLU) [20] were used along with MaxPooling. The output of the seventh *ConvLayer* was flattened and a fully connected layer was used. A drop out rate of 0.5 was used in the fully

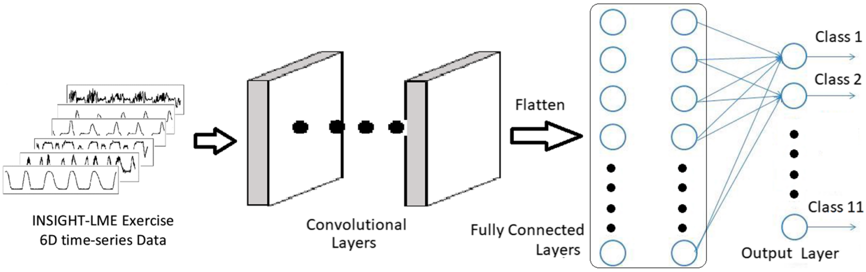


Fig. 3. CNN_Model Architecture for exercise recognition

connected layer to prevent overfitting of the data. The LME exercise recognition task was an 11 class classification problem and hence we used a fully-connected output layer with a softmax activation function capable of classifying output into 11 classes. Table 1 lists the complete list of the parameters of the CNN architecture.

The same single CNN architecture 1 was used as a binary classifier for the repetition counting task. We used a fully-connected output layer with a sigmoid activation function capable of classifying binary class. The binary class label information associated with the input was used for output prediction in the fully connected output layer. This single CNN model for repetition counting works parallel to the exercise recognition task and the predicted output are used along with exercise-type information from the exercise recognition model. Finally, a counting function was used to count the total number of repetitions using the transition information associated with the binary predicted output (Fig. 4).

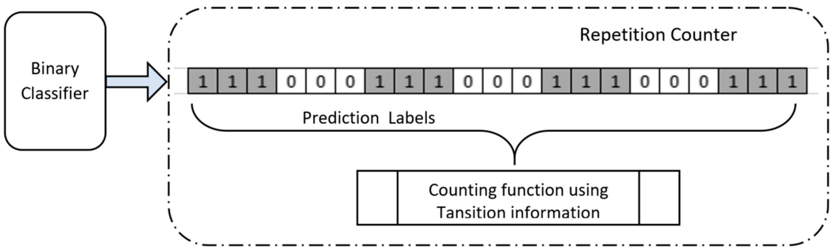


Fig. 4. Repetition Counter

The optimum model was evaluated for individual class performance based on statistical measures such as precision, recall and F1-score using Eqs. (1)–(3) respectively, where TP represents the number of times the model correctly predicts the given exercise class, FP represents the number of times the model incorrectly predicts the given exercise class and FN represents the number of times the model incorrectly predicts other than the given exercise class.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Table 1. All architecture parameters for CNN_Model2. CL: Convolution Layer and DL: Dense Layer

Layer	Value	Parameters
Input layer	2048 × 1 × 6	0
Convolution filters CL1	16	304
Kernel size CL1	(3, 1)	–
Strides CL1	(1, 1)	–
Convolution filters CL2	16	784
Kernel size CL2	(3, 1)	–
Strides CL2	(1, 1)	–
Convolution filters CL3	32	1568
Kernel size CL3	(3, 1)	–
Strides CL3	(1, 1)	–
Convolution filters CL4	32	3104
Kernel size CL4	(3, 1)	–
Strides CL4	(1, 1)	–
Convolution filters CL5	64	6208
Kernel size CL5	(3, 1)	–
Strides CL5	(1, 1)	–
Convolution filters CL6	64	12352
Kernel size CL6	(3, 1)	–
Strides CL6	(1, 1)	–
Convolution filters CL7	96	18528
Kernel size CL7	(3, 1)	–
Strides CL7	(1, 1)	–
Batch normalization CL1, CL2, CL3, CL4, CL5, CL6, CL7	Yes	64 + 64 + 128 + 128 + 256 + 256 + 384
Activation function CL1, CL2, CL3, CL4, CL5, CL6, CL7	ReLU	0
Dense Layer DL1	128	25165952
Dropout DL1	0.25	0
Dense Layer DL2	11	1419
Activation function DL2	softmax	0
Total parameters	**:**	**25,211,499**
Trainable parameters	**:**	**25,210,859**
Non-trainable parameters	**:**	**640**

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

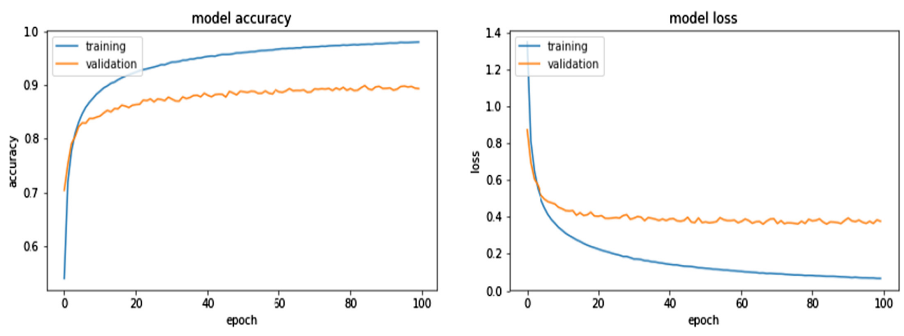
$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

4 Experimental Results

The CNN models for both tasks were constructed using Keras API [21] with the TensorFlow [22] back end with the choice of optimizer function among stochastic gradient descent (SGD) [23], Adam [24], and RMSprop [25]. The best learning rate was selected by training the model over a range of 1e−03 to 1e−10 with a decay of 1e−01. The multi-class classification model for exercise recognition was optimized using the loss functions such as categorical cross-entropy (CCE) [26] and Kullback–Leibler divergence (KLD) [27] to have lower losses. However, the binary-class model for repetition counting was optimized using binary cross-entropy loss function. We used early stopping during model building by monitoring the validation loss. A learning rate scheduler was used effectively using the "ReduceOnPlateau" function from Keras. Data augmentations like shearing, resizing, flipping, rotation were not performed on the time-series data. Models were trained using the training set and validated using the validation set. A model with a minimum validation loss and with the best validation accuracy was selected as the optimum CNN model in both tasks and was further tested using the test set.

4.1 Exercise Recognition Using CNN Model

A CNN model with an Adam optimizer having a learning rate 1e-7 and a KLD loss function was found to be the best model. The model recorded an overall training score of 96.89% and a validation score of 88.97%. For the test set, the model recorded an overall test accuracy of 95.61% and an overall F1-score measure of 96% and an overall loss of 0.1288. Figure 5(a) and Fig. 5(b) shows the learning curves in terms of training and validation accuracies as well as training and validation losses.



(a) Training and validation accuracies (b) Training and validation losses

Fig. 5. Learning curves

The performance of the CNN model, in terms of statistical parameter measurements such as precision, recall and F1-score, for individual exercise are tabulated in Table 2. The model recorded an overall precision of 96.52%, overall recall rate of 97.13% and an overall F1-score of 96.80% for the upper-body LME exercises. The overall performance for the lower body LME in terms of precision, recall rate and F1-score measures were 95.99%, 97.08% and 96.5% respectively.

Table 2. Performance evaluation measures of the CNN model

Exercise type		Precision	Recall	F1-score	Support
Upper-body LME exercises	Bicep Curls	0.9952	0.9713	0.9831	1290
	Frontal Raise	0.8917	0.9574	0.9234	1290
	Lateral Raise	0.9389	0.9178	0.9283	1290
	Triceps Extension	0.9985	1.0000	0.9992	1290
	Pec Dec	0.9953	0.9837	0.9895	1290
	Trunk Twist	0.9721	0.9977	0.9847	1290
Lower-body LME exercises	Standing Bicycle	0.9834	0.9651	0.9742	1290
	Squats	0.9874	0.9698	0.9785	1290
	Leg Lateral Raise	0.9771	0.9907	0.9838	1290
	Lunges	0.8917	0.9574	0.9234	1290
Common movements	Others	0.8975	0.8389	0.8672	1440
Micro average		0.96	0.96	0.96	14340
Macro average		0.96	0.96	0.96	14340
Weighted average		0.96	0.96	0.96	14340

4.2 Repetition Counting Using the CNN Model

The optimum model was selected based on the validation score and incorporated an Adam optimizer and had a learning rate of 1e-06. The optimum model was further tested with the test data set to count the repetitions. The test data set consisted of 30 exercise data from each exercise type corresponding to the fifteen participants performing each exercise twice and 6 to 7 repetitions over 25 s of data segment.

Table 3. Number of error counts in the repetition using CNN model

Exercise type	Acronym	Total subjects	Error count			
			$e\|0\|$	$e\|1\|$	$e\|2\|$	$e > \|2\|$
Upper-body LME exercises	BC	30	28	1	0	1
	FR	30	25	4	0	1
	LR	30	30	0	0	0
	TER	30	29	1	0	0
	PD	30	29	1	0	0
	TT	30	27	3	0	0
Lower-body LME exercises	SBC	30	18	7	4	1
	SQ	30	15	9	4	2
	LLR	30	23	5	2	0
	L	30	6	4	4	16

Table 3 shows the results of repetition counting for individual LME exercise in terms of the number of absolute errors. The total number of subjects used in the test set for testing each exercise is also indicated in the table. The repetition error counts are indicated by the columns "Error Count" or "e$|X|$", where "e$|X|$" indicates the number of exercise sets with '$|X|$' repetition error count. '$|X|$' represents the absolute error count in terms of 0, 1, 2, or more than 2 errors. The repetition counting method performed better for upper-body exercises like BC, FR, LR and TER in comparison to the repetition counting of the lower-body exercises. For example, from Table 3, for the upper-body LME exercises, zero errors in repetition counting were reported in 168 instances among 180 observed sets.

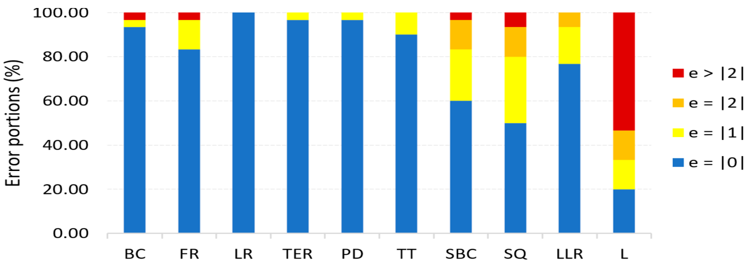


Fig. 6. Number of errors of the repetition counting using the CNN model (Color figure online)

A significant amount of error count for the upper-body LME exercises was with one count error. We could achieve 100% correct counting only in the case of LR exercise trials. Repetition counting performance for Lunges, a lower-body exercise, was very poor. Performance of the model can be evaluated with a tolerance of one repetition count error (i.e. blue + yellow, Fig. 6). The repetition counting from the model was within an error of ± 1 repetitions in 88% of the observed exercise sets.

5 Discussion

In this paper, we studied a deep CNN model architecture on the INSIGHT-LME dataset for automatic recognition and repetition counting in LME exercises. The dataset used was based on the data from single wrist-worn inertial measurement unit from the exercises used in CVD rehabilitation program. We found that the deep CNN model constructed on the time-series data was an efficient model for exercise recognition and repetition counting in terms of accuracy measure. In addition, we demonstrated a novel method of using a single model based on multi-channel time-series data for the repetition counting from all the ten exercises.

We would like to discuss the outcome of our study with the findings of recent relevant studies in the area of exercise-based rehabilitation using wearables. First, this study of ours was an extension of findings from our work [1], where a comparative approach was adopted in LME exercise recognition and repetition counting using different supervised machine learning models and a deep CNN model using AlexNet architecture. In addition, using the earlier study [1] we had made the INSIGHT-LME dataset publicly available. The CNN model using AlexNet architecture was the best approach, however, requires the input data in terms of 2D images. However, this study of exercise recognition and repetition counting uses the multi-channel raw time-series data and achieves the overall same result.

Second, Soro et al. [11], a recent work on exercise recognition and repetition counting on ten Cross-Fit exercises using deep CNN models uses two sensors one on a foot and one on hand. The study makes use of a single deep CNN model for the exercise recognition task but uses ten different models for the repetition counting. 9D data from accelerometer, gyroscope and orientation sensor was used and reports an overall accuracy measure of 97% in exercise recognition with only exercise data. In contrast, our model for the exercise recognition uses 6D data and the recognition task considers an additional eleventh class ("Others"), with non-exercise movement data along with the ten exercise class data. We built a single CNN model for repetition counting in contrast to the ten individual models.

While our studies and those of Soro et al. [11] were on different exercises and different data-sets, the main aim was to address exercise-based rehabilitation using deep learning models. The current study using multi-channel information with a deep CNN appear also shows that it is possible to use a single model to count exercise repetition, with very little loss in accuracy. This may be beneficial in reducing the dependency on the total number of resources required in repetition computation in the case of multiple exercise evaluation.

6 Conclusion

We studied a single deep CNN architecture based model on the exercises used in an exercise-based CVD rehabilitation program. The automatic recognition and repetition counting of the exercises was achieved using multi-channel (6D) time-series data obtained from a single wearable sensor. We achieved an overall F1-score measure of 96% in the exercise recognition task and the repetition counting was within an error of ± 1 count among 88% of the observed exercise sets. Our study also showed that it is possible to use a single CNN model for repetition count with very little loss in accuracy.

Acknowledgement. This publication has emanated from research supported by INSIGHT SFI Research Centre for Data Analytics and Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund.

References

1. Prabhu, G., O'Connor, N.E., Moran, K.: Recognition and repetition counting for local muscular endurance exercises in exercise-based rehabilitation: a comparative study using artificial intelligence models. Sensors **20**(17), 4791 (2020)
2. Buys, R., et al.: Cardiac patients show high interest in technology enabled cardiovascular rehabilitation. BMC Med. Inform. Decis. Mak. **16**(1), 95 (2016)
3. Dalal, H.M., Zawada, A., Jolly, K., Moxham, T., Taylor, R.S.: Home based versus centre based cardiac rehabilitation: cochrane systematic review and meta-analysis. BMJ **340**, b5631 (2010)
4. Foerster, F., Smeja, M., Fahrenberg, J.: Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. Comput. Hum. Behav. **15**(5), 571–583 (1999)
5. Lara, O.D., Labrador, M.A.: A survey on human activity recognition using wearable sensors. IEEE Commun. Surv. Tutor. **15**(3), 1192–1209 (2012)
6. Weiss, G.M., Timko, J.L., Gallagher, C.M., Yoneda, K., Schreiber, A.J.: Smartwatch-based activity recognition: a machine learning approach. In: 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 426–429. IEEE (2016)
7. Sarcevic, P., Pletl, S., Kincses, Z.: Comparison of time-and frequency-domain features for movement classification using data from wrist-worn sensors. In: 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), pp. 000261–000266. IEEE (2017)
8. Chernbumroong, S., Atkins, A.S., Yu, H.: Activity classification using a single wrist-worn accelerometer. In: 2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings, pp. 1–6. IEEE (2011)
9. Zhu, C., Sheng, W.: Recognizing human daily activity using a single inertial sensor. In: 2010 8th World Congress on Intelligent Control and Automation, pp. 282–287. IEEE (2010)
10. Crema, C., Depari, A., Flammini, A., Sisinni, E., Haslwanter, T., Salzmann, S.: IMU-based solution for automatic detection and classification of exercises in the fitness scenario. In: 2017 IEEE Sensors Applications Symposium (SAS), pp. 1–6. IEEE (2017)
11. Soro, A., Brunner, G., Tanner, S., Wattenhofer, R.: Recognition and repetition counting for complex physical exercises with deep learning. Sensors **19**(3), 714 (2019)
12. Um, T.T., Babakeshizadeh, V., Kulić, D.: Exercise motion classification from large-scale wearable sensor data using convolutional neural networks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2385–2390. IEEE (2017)
13. O'Reilly, M.A., Whelan, D.F., Ward, T.E., Delahunt, E., Caulfield, B.: Classification of lunge biomechanics with multiple and individual inertial measurement units. Sports Biomech. **16**(3), 342–360 (2017)
14. Whelan, D., O'Reilly, M., Ward, T., Delahunt, E., Caulfield, B.: Evaluating performance of the single leg squat exercise with a single inertial measurement unit. In: Proceedings of the 3rd 2015 Workshop on ICTs for Improving Patients Rehabilitation Research Techniques, pp. 144–147 (2015)
15. Morris, D., Saponas, T.S., Guillory, A., Kelner, I.: RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3225–3234 (2014)

16. Mortazavi, B.J., Pourhomayoun, M., Alsheikh, G., Alshurafa, N., Lee, S.I., Sarrafzadeh, M.: Determining the single best axis for exercise repetition recognition and counting on smartwatches. In: 2014 11th International Conference on Wearable and Implantable Body Sensor Networks, pp. 33–38. IEEE (2014)
17. Whelan, D., O'Reilly, M., Ward, T., Delahunt, E., Caulfield, B.: Evaluating performance of the lunge exercise with multiple and individual inertial measurement units. In: Pervasive Health 2016: 10th EAI International Conference on Pervasive Computing Technologies for Healthcare, Cancun, Mexico, 16–19 May 2016. ACM (2016)
18. Prabhu, G., Ahmadi, A., O'Connor, N.E., Moran, K.: Activity recognition of local muscular endurance (LME) exercises using an inertial sensor. In: Lames, M., Saupe, D., Wiemeyer, J. (eds.) IACSS 2017. AISC, vol. 663, pp. 35–47. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67846-7_4
19. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
20. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: ICML (2010)
21. Chollet, F., et al.: Keras (2015). https://github.com/fchollet/keras
22. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/. Software available from tensorflow.org
23. Kiefer, J., Wolfowitz, J., et al.: Stochastic estimation of the maximum of a regression function. Ann. Math. Stat. **23**(3), 462–466 (1952)
24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
25. Tieleman, T., Hinton, G.: Lecture 6.5–RmsProp: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw. Mach. Learn. **4**, 26–31 (2012)
26. Rubinstein, R.Y., Kroese, D.P.: The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning. Springer, New York (2004). https://doi.org/10.1007/978-1-4757-4321-0
27. Joyce, J.M.: Kullback-Leibler divergence (2011)