

Activity recognition in a Physical Interactive RoboGame

Ewerton L.S Oliveira*, Davide Orrù*, Tiago Nascimento† and Andrea Bonarini*

*Dipartimento di Elettronica, Informazione e Bioingegneria - AI and Robotics Lab. Politecnico di Milano, Italy.

†Universidade Federal da Paraíba, João Pessoa, Brazil.

Abstract—In this paper, we investigate the possibility of human physical activity recognition in a robot game scenario. Being able to recognize types of activity is essential to enable robot behavior adaptation to support player engagement. Also, the introduction of this recognition system will allow for development of better models for prediction, planning and problem solving in PIRGs that can foster human-robot interaction. The experiments reported on this paper were performed on data collected from real in-game activity, where a human player faces a mobile robot. We use a custom single tri-axial accelerometer module attached to the player’s chest in order to capture motion information. The main characteristic of our approach is the extraction of features from patterns found on the motion variance rather than on raw data. Furthermore, we allow for the recognition of unconstrained motion given that we do not ask the players to perform target activities before hand: all detectable activities are derived from the free player motion during the game itself. To the best of our knowledge, this is the first paper to consider activity recognition in a physical interactive robogame.

I. INTRODUCTION

Detailed measurement and classification of an individual’s physical activity is fundamental in order to understand the relationship between physical activity and health but, also, to achieve an enhanced level of interaction between humans and robots. In particular when considering Physically Interactive RoboGames (PIRG [1]) – where the objective is the exploitation of both the real world as environment, and of one or more real, physical, autonomous robots as game opponents or companions of human players – activity recognition plays a fundamental role to adapt the robot strategy to support the player’s entertainment during the game.

In this paper, we propose a model which aims at quantifying and classifying the player’s activity in a PIRG game using a 3-axis custom accelerometer positioned on player’s chest. We define a set of high level activity classes that are automatically classified relying on a supervised machine learning framework that considers the changes in the raw acceleration signal.

We propose a mathematical model to quantify the general amount of activity, also taking into account the type of the activity that has been performed so far.

The paper is organized as follows: we first present some related works about activity recognition and classification; then, we describe the game scenario on which we carried out our experiment and the relative human player and robotic platform roles. Sections III and IV explain how we collected data, while Section V provides the description of the activity model. The results are finally discussed in section VI.

II. RELATED WORKS

A number of recent studies have investigated activity recognition using one or more accelerometer placed in different part of the body. [2] proposes daily physical activities recognition using a single accelerometer placed at the wrist or ankle. This study uses various combinations of window lengths and feature sets to develop an algorithm with the aim to classify a wide variety of activities into four distinct classes using accelerometer raw data and, also, higher classification accuracies for ankle data w.r.t the wrist data.

The possibility of performing human activity recognition using raw accelerometer data from smart-phones is also a very common field of study. For example, in [3] an experiment where subjects were requested to perform two different activities has been performed and, given their start and end times, two datasets were considered: one for in-hand phone position and one for in-pocket phone position. This information was then used to label different activities for a supervised learning classification problem. Another similar work was [4], while in [5] a number of applications for such studies are also proposed.

Considering a game environment, [6] investigated sensor placement and modality for activity recognition within the context of childrens playground activities. By mean of parallel sensing, performed using a set of smart-phones, activity dependent data have been generated. The obtained set of data was then used to train decision tree classifiers. This study shows once again that phones placed closer to the core of the body generate better models than phones placed on the extremities.

Similarly, in [7] a stochastic approximation framework for intensity-independent activity recognition based on clustering techniques is proposed. The aim is to enhance and automate the calculation of metabolic equivalent of task (MET) and also to improve an exergaming (video games that are also a form of exercise) platform consisting of two main components: an accelerometer-embedded belt and an RPG video game called FreedroidRPG that was used as an incentive for the participant to perform physical activity throughout the day. The study shows the ability of the used stochastic approximation framework to extrapolate unknown intensity levels from a few known intensity levels that can be used to enhance activity recognition.

Several studies in literature also focused on the comparison between multi-sensor versus single-sensor activity detection

and also on the optimal body placement of such sensors.

In [8] is reported the evaluation of accelerometer-based multi-sensor versus single-sensor activity recognition systems. The objective is to compare two distinct types of wearable systems: single-sensor wearable systems adopting complex algorithms and multi-sensor systems employing light-weight algorithms. The impact of the sampling rate on the recognition accuracy was then investigated using four classifiers. The experimental results illustrated that the recognition accuracy was steady at 50-Hz and above, and the single sensor system was more sensitive to the sampling rate than the multi-sensor system.

The work of [9] is, in turn, focused on making a comparison between the activity recognition rates of an activity classifier trained on acceleration signal collected on the wrist and hip. During the experiments 52 children and adolescents completed 12 activity trials that were categorized into 7 activity classes: lying down, sitting, standing, walking, running, basketball, and dancing. As result, the hip model exhibited excellent classification accuracy for sitting, standing, walking, and running; acceptable classification accuracy for lying down and basketball; and modest accuracy for dance. The wrist model exhibited excellent classification accuracy for sitting, standing, and walking; acceptable classification accuracy for basketball; and modest accuracy for running, lying down and dance.

We are presenting an activity recognition method based on the variation of the accelerometer signal instead of relying on raw data. We also perform the activity recognition during a real RoboGame scenario, letting the human player free to move unconstrained in the playground without asking to perform pre-defined target movements.

III. THE GAME SCENARIO

In order to test our model, we designed a competitive game, where the human player faces an autonomous robot.

A. The Robotic Platform

We have adopted a holonomic robot that is free to move in any direction at a maximum speed comparable to that of people in indoor environments (1.4 m/sec).

The robot body consists of a robust, triangular, omnidirectional base 5cm high (with 40 cm in diameter), where motors, batteries and electronics are embedded. On the base, an auxiliary aluminum structure is mounted, having the on-board computer attached laterally and a Kinect sensor on top. In total, the robot is 85cm high (see figure 1(a)).

B. Game Scenario

For the present work, the “playground” is a rectangular area of 4m×2m where, on each corner, “towers” are placed. Each tower is equipped with a button and four LEDs that can be progressively turned on, one by one, by pressing the button on the tower top. Each LED requires the button to be pressed for about 2.5 seconds, meaning that the tower takes about 10 seconds of button push in order to light up all of the four LEDs. Button pressing time is cumulated and can be

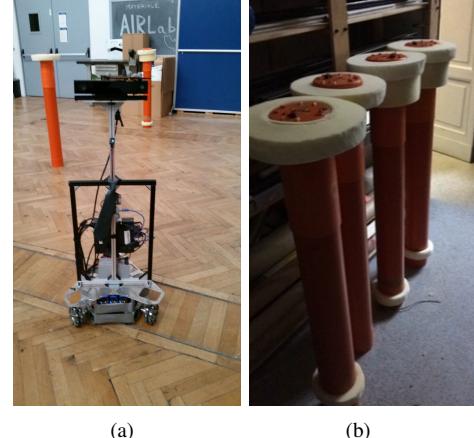


Fig. 1: a) the robotic platform and b) the target tower used in the game (height 110cm).

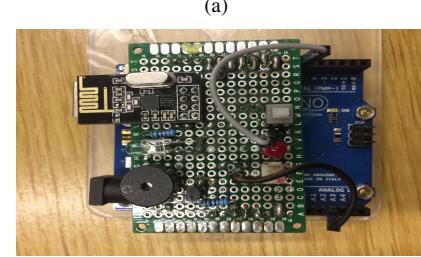
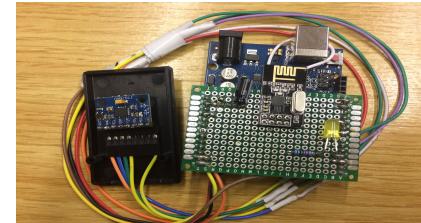


Fig. 2: a) the accelerometer circuit and b) the tower circuit used in the game, both based on Arduino boards.

distributed on different moments. After turning on all LEDs of a particular tower during game play, it is said that the player has captured it.

In order to win, the human player must be able to secure all the existing towers without letting a single one be ruined down by the robot. If, at anytime, a tower falls (because of the robot or player) the game ends and the human player is defeated.

The robot is a holonomic base (see Figure 1(a)), it is able to move across the entire playground just as the human player and is only constrained by the fact that an already captured tower, or one whose button is currently pressed by the player cannot be teared down. The player can also block the robot path by staying in front of it. Notice that while the player is trying to capture a given tower, the robot can try to put down another one.

The robot must avoid hitting the player at any time during the game. Figure 1(b) presents the towers that were used in the game.

This game is designed so that the robot, given its agility and maximum speed (1.4 m/sec) could always win, but it should show the appropriate and believable behavior to keep the player engaged and interested.

C. The Human Player

When playing the game, we ask the human player to wear a colored robe (see figure3) in order to allow for blob detection and tracking, leading to feature extraction. The visual features extracted in the game are out of the scope of this paper, since we describe only results relying on the accelerometer data.

To the scope of this paper, we have used a custom accelerometer board attached to the player's chest in order to capture a detailed player motion information. The choice of the accelerometer position was conditioned by the need of avoiding the introduction of noise from irrelevant player motion. With the sensor placed on the chest we aimed at getting a more stable reference for motion in comparison to, for instance, data gathered from the sensor placed on the arm or thigh.

The custom accelerometer module is based on the InvenSense MPU-6050 3-axis accelerometer board and an Arduino Uno micro-controller. The circuit also contains a Nrf24l01 radio-frequency module that allows the accelerometer data to be sent to the on-board computer.

IV. DATA COLLECTION

For this work, we considered 29 matches involving 15 male participants of different ages. The age distribution consisted of children (7-10) and adults (26-40). Matches had a minimum time duration of about 40 seconds and a maximum of about 1 minute and 10 seconds.

The collected data correspond to acceleration values along x, y, and z axis with a sampling frequency of 50Hz, which is five times larger than the frequency considered to be sufficient for detecting daily activities from accelerometer data (10Hz) [10], [11], [12].



Fig. 3: Human player (in magenta) during the game. The playground configuration consisted of 3 target towers.

V. ACTIVITY ANALYSIS

We inspected the logged data in order to identify recurrent movements that compose a dictionary of activities. From the accelerometer data, we were able to identify a few high-level activity types, such as the one listed in table I.

TABLE I: List of high-level activity classes.

Class	Description
running	describes a running activity. For this paper, multiple styles of running are not considered. For instance, “fast” or “slow” running are considered the same.
walking/dodging	represents the walking and dodging activity. The latter refers to a sudden quick movement to avoid the robot or to call its attention.
locally_moving	a player generic motion that is too small to fall into other categories, but not so small to be characterized as inactivity. Robot path-blocking motions also fall into this type of motion.
inactive	motion that are too low in intensity to be characterized as one of the above activities. A crisp threshold is used to delimit this category.

Additionally, we identified that an activity is often followed by a short period of inactivity, i.e., a short period where the player is relatively inactive. Those observed periods are a characteristic that comes from the fact that the game naturally enforces such behavior when the player is, for instance, pushing a button on a given tower or trying to block the robot's path. On these moments, the changes in acceleration are usually small, thus it is possible to see a relative flatness of the signal (secs 9-12 on Figure 5). Furthermore, it is possible to see a difference between activities in figure 7 and figure 8.

Nonetheless, one may claim that such inactivity periods are a common characteristic present in any physical game, both related to a resting moment following an intense period of activity or even periods where the player is waiting to move in a favorable moment.

Since, by means of an accelerometer, motion is associated to a considerable amount of turbulence in the signal, we exploit the signal standard deviation as a measure of such turbulence as well as a way to emphasize motion patterns located between the periods of inactivity.

In this paper, the computation of the the standard deviation over time is done by a sliding window method, which results in the generation of a continuous graph as shown on Figure 5. Working this way, turns out to be simpler than to perform data annotation by, for instance, relying on a fixed size sliding windows method. In our case, activities, such as “running”, do not have fixed time duration, which makes the applicability of fixed sliding windows methods not well suitable [13].

To characterize the inactive type of activity, we rely on the definition of a threshold that precisely categorizes the class. If the STD is below that threshold, the system purely identifies the motion as of “inactive” type. This class mainly represents player activity that is too small to be of any interest.

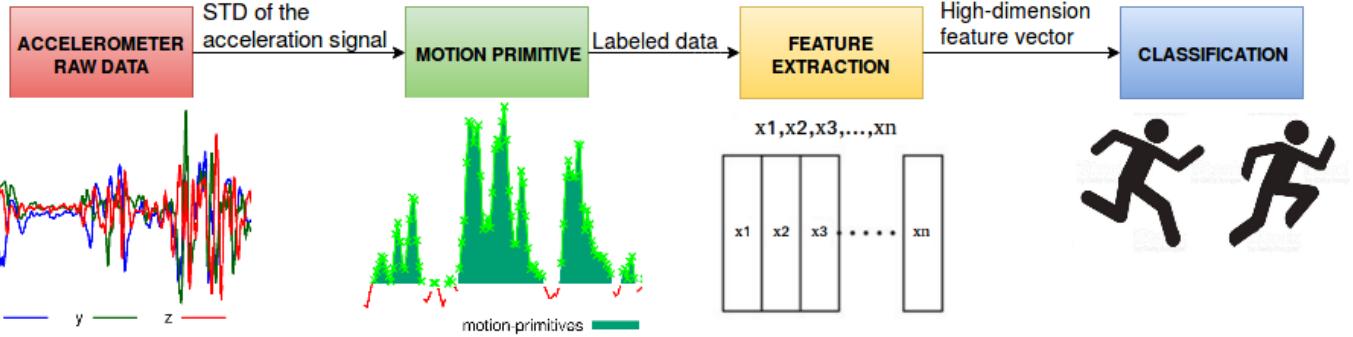


Fig. 4: Overview of the activity recognition system.

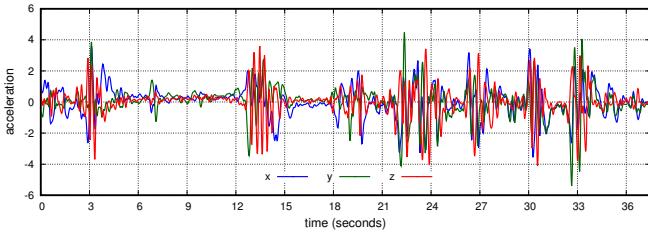


Fig. 5: Graph of acceleration in x, y and z axis for a game that lasted about 40 seconds.

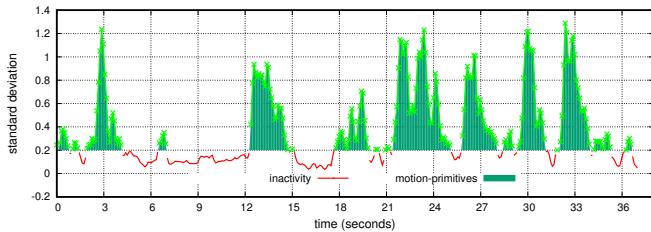


Fig. 6: Standard deviation of the acceleration in Figure 5, computed using a sliding window of half a second. The red line portions represent variance values inside the inactivity zone (below a threshold of 0.2). Green areas are referenced as “motion primitives”.

When performing data annotation, we do not consider “inactive” as an activity instance in the dataset, since the threshold is used to fully determine it. However, we pay attention to each STD data interval above the threshold, which we call a “motion primitive” (Figure 6). The motion primitives are the data aggregations to which we associate a class label in the annotation procedure.

It is important to recall, however, that the annotated data come from real game interaction between a human and a mobile robot, thus, we do not limit the motion of the participants by asking them to perform the target activities (Table I). This differs our work from those reported in popular papers in the Activity Recognition community, such as [4], [14].

For the classification, we have extracted the following descriptors from the motion primitives (recall, they refer to standard deviation of the raw values):

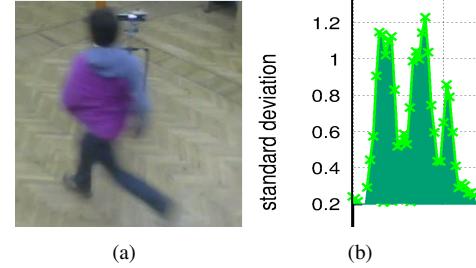


Fig. 7: (a) Player performing a running activity. (b) Associated running motion primitive.

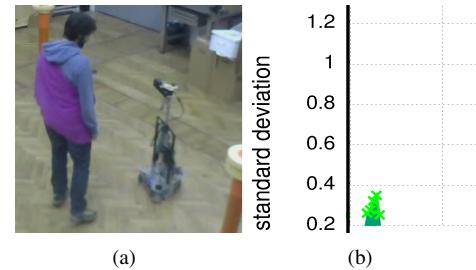


Fig. 8: (a) Player performing local movements. (b) Associated motion primitive.

- **mean**: the mean values.
- **activity_time**: the time duration in seconds.
- **max_peaks**: the max peak.
- **number_of_peaks**: the number of peaks.
- **mean_of_peaks**: the mean of all peaks.
- **max-min**: difference between max and min.
- **std**: standard deviation.
- **mad**: median absolute deviation.
- **sma**: signal magnitude area.
- **energy**: the signal energy.
- **iqr**: interquartile range.
- **mean_over_max**: mean of peaks divided by the max peak.
- **maxInd**: index of the frequency component with largest magnitude.
- **meanFreq**: weighted average of the frequency components to obtain a mean frequency.

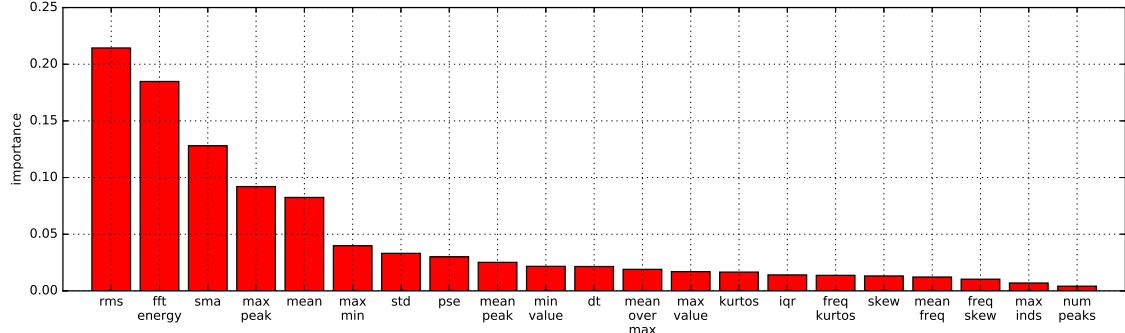


Fig. 9: Feature importance computed by a forest of decision trees classifiers.

- **skewness:** skewness of the motion primitive.
- **kurtosis:** kurtosis of the motion primitive.
- **freq-skewness:** skewness of the frequency domain signal.
- **freq-kurtosis:** kurtosis of the frequency domain signal.
- **pse:** Power spectral entropy.
- **rms:** Return the root mean square.

A holistic view of the classification approach is detailed in Figure 4.

VI. RESULTS AND DISCUSSION

In this section, we describe our experiments and then present and discuss the results obtained with our methodology.

A. Description of Experiments

As pointed out, first we build the standard deviation graph from the raw accelerometer data and then collect the labeled motion primitives. The resulting dataset is described in Table II and is subsequently used for training and testing the activity classifiers.

The STD graph was generated considering a sliding window 500msec long. This time length turned out to be descriptive enough to produce variance intervals that made it possible to distinguish activities.

As argued above, the “inactive” type is not considered in the recognition task since it is directly classified by the inactive threshold analysis.

TABLE II: Motion primitive Dataset

Id	Motion primitive	#	%
1	locally moving (LM)	123	34
2	walking/dodging (WD)	93	25
3	running (R)	151	41

Before training any classifier, we performed feature selection by evaluating the importance of features using random forest method, composed by 300 decision trees (see Figure 9).

We tested different classifiers using 10-fold cross validation in order to have a more descriptive accuracy information. Following common practice, the train-test dataset ratio was defined as 80% and 20% respectively.

TABLE III: Cross-validation accuracy results for several classification methods using the 5 most significant features show in Figure 9.

Method	Accuracy
SVM (Linear Kernel)	0.80 (+/- 0.08)
Random Forest	0.81 (+/- 0.06)
Gaussian Naive Bayes	0.80 (+/- 0.11)
Ensemble (Hard voting)	0.82 (+/- 0.10)
AdaBoost	0.65 (+/- 0.40)

TABLE IV: Classification report for the chosen ensemble method (Random Forest)

	precision	recall	f1-score	support
LM	0.88	0.91	0.89	23
WD	0.79	0.60	0.68	25
R	0.77	0.92	0.84	26
avg/total	0.81	0.81	0.80	74

Table III presents 10-fold cross validation results using different classifiers on the five most important features, that is: *rms*, *fft_energy*, *sma*, *max_peak* and *mean*.

The ensemble in Table III is defined as a majority (Hard) voting approach by the combination of the SVM, Gaussian Naive Bayes and Random Forest. The AdaBoost method, in turn, takes a combination of 100 weak classifiers (Decision Trees). All methods were trained by using Python Scikit-learn machine learning library.

Despite the effort, with a confidence interval of 95% we see that SVM, Random Forest, Gaussian Naive Bayes as well as their ensemble have a similar accuracy result. By considering the variance in their result, we see that Random Forest gives the most stable result.

Given the 10-fold cross validation results, we decided to use as final method the Random Forest ensemble classifier (10 decision trees). Detailed results for the method are shown in table IV and the corresponding confusion matrix and Receiver Operating Characteristic in Figure 10. The majority of mistakes in the classification correspond to the difficulty in separating “walking/dodging” from a “running” activity. This is most probably caused by similarities in acceleration levels produced by the dodging activity.

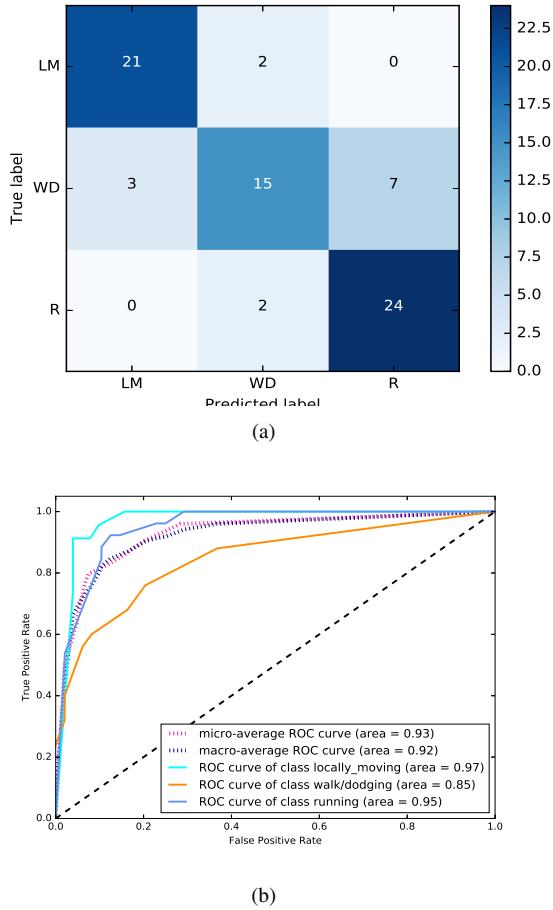


Fig. 10: a) confusion matrix for the trained Random Forest ensemble method and the associated b) ROC curve.

VII. CONCLUSION

In this work, we have investigated the recognition of high-level human player activity in a Physically Interactive RobotGame (PIRG) scenario, where a human player faces a mobile robot. We have used the variance in player motion as data instances and primary source of information from where to extract features and then train a machine learning model.

Physically Interactive RoboGames are a rather new style of game and provide a specific setting from which to study human-robot interaction in situations framed by rules.

In order to design better PIRGs where, for instance, robots can adapt their strategies to support the player entertainment, a player activity model is desired, for which activity recognition is fundamental.

We are currently working on the real-time applicability of this method as well as its validity for supporting a full player engagement model, based on the mixture of activity and interaction level.

ACKNOWLEDGMENT

This work was in part supported by the Brazilian National Council for Scientific and Technical Development (CNPq)

under the Science Without Border Scholarship Program.

REFERENCES

- [1] D. Martinia, D. Calandriello, and A. Bonarini, "Physically interactive robogames: Definition and design guidelines," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 739–748, 2013.
- [2] A. Mannini, S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell, "Activity recognition using a single accelerometer placed at the wrist or ankle," *Medicine and science in sports and exercise*, vol. 45, no. 11, p. 2193, 2013.
- [3] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Computer Science*, vol. 34, pp. 450–457, 2014.
- [4] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *ESANN*, 2013.
- [5] X. Su, H. Tong, and P. Ji, "Activity recognition with smartphone sensors," *Tsinghua Science and Technology*, vol. 19, no. 3, pp. 235–249, 2014.
- [6] N. Jablonsky, S. McKenzie, S. Bangay, and T. Wilkin, "Evaluating sensor placement and modality for activity recognition in active games," in *Proceedings of the Australasian Computer Science Week Multiconference*. ACM, 2017, p. 61.
- [7] N. Alshurafa, W. Xu, J. J. Liu, M.-C. Huang, B. Mortazavi, C. K. Roberts, and M. Sarrafzadeh, "Designing a robust activity recognition framework for health and exergaming using wearable sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1636–1646, 2014.
- [8] L. Gao, A. Bourke, and J. Nelson, "Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems," *Medical engineering & physics*, vol. 36, no. 6, pp. 779–785, 2014.
- [9] S. G. Trost, Y. Zheng, and W.-K. Wong, "Machine learning for activity recognition: hip versus wrist data," *Physiological measurement*, vol. 35, no. 11, p. 2183, 2014.
- [10] L. Atallah, B. Lo, R. King, and G.-Z. Yang, "Sensor placement for activity detection using wearable accelerometers," in *Body Sensor Networks (BSN), 2010 International Conference on*. IEEE, 2010, pp. 24–29.
- [11] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Aaai*, vol. 5, no. 2005, 2005, pp. 1541–1546.
- [12] B. Kikhia, M. Gomez, L. L. Jiménez, J. Hallberg, N. Karvonen, and K. Synnes, "Analyzing body movements within the laban effort framework using a single accelerometer," *Sensors*, vol. 14, no. 3, pp. 5725–5741, 2014.
- [13] M. H. M. Noor, Z. Salcic, I. Kevin, and K. Wang, "Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer," *Pervasive and Mobile Computing*, 2016.
- [14] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.