

Article

# Activities of Daily Living and Environment Recognition Using Mobile Devices: A Comparative Study

José M. Ferreira <sup>1,†</sup>, Ivan Miguel Pires <sup>2,3,\*,†</sup> , Gonçalo Marques <sup>2,†</sup> , Nuno M. Garcia <sup>2,†</sup> , Eftim Zdravevski <sup>4,†</sup> , Petre Lameski <sup>4,†</sup>, Francisco Flórez-Revuelta <sup>5,†</sup> and Susanna Spinsante <sup>6,†</sup> and Lina Xu <sup>7,†</sup> 

<sup>1</sup> Computer Science Department, University of Beira Interior, 6200-001 Covilha, Portugal; jose.ferreira@ubi.pt

<sup>2</sup> Institute of Telecommunications, University of Beira Interior, 6200-001 Covilha, Portugal; goncalosantosmarques@gmail.com (G.M.); ngarcia@di.ubi.pt (N.M.G.)

<sup>3</sup> Computer Science Department, Polytechnic Institute of Viseu, 3504-510 Viseu, Portugal

<sup>4</sup> Faculty of Computer Science and Engineering, University Ss Cyril and Methodius, 1000 Skopje, Macedonia; eftim.zdravevski@finki.ukim.mk (E.Z.); petre.lameski@finki.ukim.mk (P.L.)

<sup>5</sup> Department of Computing Technology, University of Alicante, P.O. Box 99, E-03080 Alicante, Spain; francisco.florez@ua.es

<sup>6</sup> Department of Information Engineering, Marche Polytechnic University, 60131 Ancona, Italy; s.spinsante@staff.univpm.it

<sup>7</sup> School of Computer Science, University College Dublin, Dublin 4, Ireland; lina.xu@ucd.ie

\* Correspondence: impires@it.ubi.pt; Tel.: +351-966-379-785

† These authors contributed equally to this work.

Received: 18 December 2019; Accepted: 14 January 2020; Published: 18 January 2020



**Abstract:** The recognition of Activities of Daily Living (ADL) using the sensors available in off-the-shelf mobile devices with high accuracy is significant for the development of their framework. Previously, a framework that comprehends data acquisition, data processing, data cleaning, feature extraction, data fusion, and data classification was proposed. However, the results may be improved with the implementation of other methods. Similar to the initial proposal of the framework, this paper proposes the recognition of eight ADL, e.g., walking, running, standing, going upstairs, going downstairs, driving, sleeping, and watching television, and nine environments, e.g., bar, hall, kitchen, library, street, bedroom, living room, gym, and classroom, but using the Instance Based k-nearest neighbour (IBk) and AdaBoost methods as well. The primary purpose of this paper is to find the best machine learning method for ADL and environment recognition. The results obtained show that IBk and AdaBoost reported better results, with complex data than the deep neural network methods.

**Keywords:** activities of daily living; AdaBoost; mobile devices; artificial neural networks; deep neural networks

## 1. Introduction

The use of mobile devices while doing daily activities is increasing [1]. These devices have different types of sensors that allow the acquisition of several data related to the user, including the accelerometer, magnetometer, gyroscope, Global Positioning System (GPS) receiver, and microphone [2,3]. These sensors allow the creation of intelligent systems to improve the quality of life. The monitoring of older adults or people with chronic diseases is one of the critical purposes. Furthermore, it can be useful to support sports activities and stimulate the practice of physical activity in teenagers [4]. The development of these systems is included in the research of Ambient Assisted Living (AAL) systems and Enhanced Living Environments (ELE) [5–10].

The automatic recognition of ADL is widely researched [11–16], where the previously proposed framework [2,17–25] was tested and validated with different types of Artificial Neural Networks (ANN) [26–28], verifying that the best results were achieved with Deep Neural Networks (DNN). The proposed framework allows the recognition of eight ADL, i.e., walking, running, standing, going upstairs, going downstairs, watching television, sleeping, driving, and other activities without motion, and nine environments, i.e., bar, classroom, gym, hall, kitchen, library, street, bedroom, and living room. This framework uses sensors available in mobile devices [29,30], reporting different accuracies. The proposed architecture is composed of data acquisition, data processing, data fusion, and data classification. The classification module is divided into three small stages, including the recognition of simple ADL, i.e., running, standing, walking, going upstairs, going downstairs, and other activities without motion, with accelerometer, gyroscope, and magnetometer sensors, the recognition of environments, i.e., bar, classroom, gym, hall, kitchen, library, street, bedroom, and living room, with the microphone data, and the recognition of activities without motion, i.e., sleeping, watching television, driving, and other activities without movement.

This research is based on the creation of a framework for the recognition of ADL and its environments. Still, its main goal is related to the testing of ensemble learning methods to further improve the obtained accuracy in the recognition.

The main contribution of this paper is the implementation of different machine learning methods with the same dataset used for the creation of the framework [31], including AdaBoost [32,33] and Instance Based k-nearest neighbour (IBk) [34], using different Java based frameworks, including Weka [35] and Smile [36]. Finally, the results obtained with the different methods should be compared to decide the best method for implementation using the ADL and environment recognition framework.

The results show that the application of the IBk method implemented with Weka software reported better results than others, reporting results with around 77.68% accuracy in recognition of ADL, 41.43% accuracy in recognition of environments, and 99.73% accuracy in recognition of activities without motion. However, AdaBoost applied with Smile also gave important results, reporting results between 85.44% (going upstairs) and 99.98% (driving).

Section 2 gives the presentation of the different methods implemented. The results and the comparative study of this paper are presented in Section 3. Finally, the discussion and conclusions are presented in Section 4.

## 2. Methods

### 2.1. Study Design

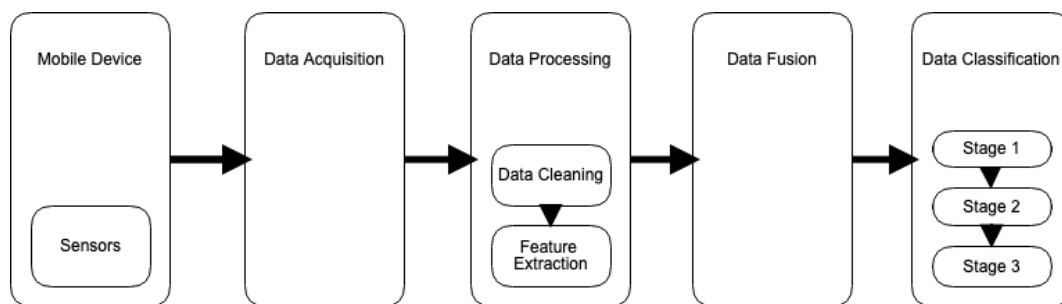
This study consisted of the use of the same structure and data acquired by the research presented in [18,21,22,24,25] to implement a comparative study between three types of studies. The tests were conducted with the dataset available in [24], which included data related to the eight ADL and nine environments. The information was acquired from the accelerometer, magnetometer, gyroscope, microphone, and GPS receiver available in the mobile device.

As presented in [21], an Android application was used for the acquisition of the data related to the different sensors. This mobile application is responsible for data acquisition and data processing using built-in smartphone sensors such as the accelerometer, magnetometer, gyroscope, sound, and GPS data. The software was responsible for managing five seconds of data every five minutes. It was installed in a smartphone, and it was placed in the front pocket of the pants of 25 subjects with different lifestyles, aged between 16 and 60 years old. For ADL and environment identification, a minimum of 2000 samples with five seconds of data acquired from the different sensors was available in the dataset used for this research. Different environments were used in the performed tests and were strictly related to specific activities. The volunteers had to select the ADL that would be performed using the mobile application before the start of the test. By default, the mobile application did not save any data without user input. However, the proposed method had limitations related to battery consumption

and the processing power needed to perform the tests. Currently, the majority of the smartphones available on the market incorporate high performance processing units that can be used to perform the tests, and the main problem is related to power consumption. However, most people usually recharge their mobile phones daily. Therefore, the proposed method can be used in real-life scenarios.

## 2.2. Overview of the Framework for the Recognition of the Activities of Daily Living and Environments

Based on the previously proposed framework [20], Figure 1 shows a framework composed of four stages, including data acquisition, data processing, data fusion, and data classification. The data processing consisted of several phases, including data cleaning and feature extraction. The data classification was divided into three stages, the recognition of simple ADL (Stage 1), the identification of environments (Stage 2), and the activities without motion (Stage 3). Stage 1 included the use of the data acquired from the accelerometer, magnetometer, and gyroscope sensors. The data received from the microphone were processed in Stage 2. Finally, Stage 3 increased the number of sensors, combining the data acquired from the accelerometer, magnetometer, and gyroscope sensors with the data obtained from the GPS receiver and the environment previously recognised.



**Figure 1.** Flowchart of the ADL and environment recognition framework implemented in this study.

Mobile devices are composed of several sensors, which are capable of acquiring different types of data. The framework proposed was capable of acquiring and analysing 5 seconds of data and identifying the current ADL executed and the current environment frequented. The next stage consisted of the processing of the data acquired from the sensors for a further fusion of the different data acquired from the sensors. The final module of the framework consisted of the classification of the data, which started to process all features extracted from the sensors available in the mobile device and identified if the ADL executed was available in the set of ADL proposed. In the affirmative case, the ADL performed was presented to the user. Next, the environment frequented was recognised in the next stage, and it was presented to the user. If no ADL was recognised or the ADL recognized was standing, the identification of a standing ADL would be executed, trying to discover the activity performed by the user.

### 2.2.1. Data Acquisition

This study was based on the same dataset used in [21], which is publicly available in [31]. This dataset was composed of small sets of data (five seconds every five minutes) captured by the sensors available in the off-the-shelf mobile phones, i.e., accelerometer, magnetometer, gyroscope, microphone, and GPS receiver, and stored in the cloud. The dataset used in the presented study was created using an Android mobile application for data collection. On the one hand, the running and walking data were collected in outdoor environments. On the other hand, standing and going down and upstairs were performed inside buildings.

Moreover, the tests were conducted at different times of the day. In total, thirty-six hours of data were collected, which corresponded to 2000 samples with five seconds of raw sensor data each. Before data acquisition, the user had to use the smartphone to select the ADL that would be conducted and the time needed.

### 2.2.2. Data Cleaning

Data cleaning is a step performed during data processing. It is mainly used to minimise the effects of the environmental noise acquired during the acquisition of the data from the sensors. Data cleaning methods depend on the type of data acquired and the sensors used. On the one hand, a low pass filter was applied to the data obtained from the accelerometer, magnetometer, and gyroscope sensors [37]. On the other hand, the Fast Fourier Transform (FFT) [38] was used to extract the relevant information from the data collected from the microphone. There were no methods needed to clean the received data from the other types of sensors.

### 2.2.3. Feature Extraction

After the cleaning of the data, we extracted the features. Table 1 presents the extracted features from the selected sensors, which consisted mainly of statistical features. In Stage 1, the statistical features were mainly used, i.e., standard deviation, mean, maximum and minimum value, variance, and median, of the raw data and the peaks of the motion and magnetic sensors. It also included the calculation of the five greatest distances between calculated peaks. Stage 2 was composed of the feature acquired from the microphone, including the statistical features, i.e., standard deviation, mean, maximum and minimum value, variance, and median, of the raw data, and the calculation of 25 Mel frequency cepstrum coefficients with the microphone. Finally, Stage 3 included also the distance travelled calculated from the Global Positioning (GPS) receiver data and the environment recognised in Stage 2.

**Table 1.** Features extracted.

Sensor	Type of Data	Features
Accelerometer Magnetometer Gyroscope	Raw data	standard deviation, mean, maximum and minimum value, variance, and median
	Peaks	five greatest distances between peaks, mean, standard deviation, variance, and median
Microphone	Raw data	26 MFCC, standard deviation, mean, maximum value, minimum value, variance, and median
GPS receiver	Raw data	distance travelled

### 2.2.4. Data Fusion and Classification

Data fusion and classification were included in the last stage of the ADL and environment recognition framework. The previous studies reported that the best accuracies were achieved with the DNN method [18,21,22,24,25], and all the features are presented in Table 1. This study presents the results of the test and validation of different methods, including IBk, AdaBoost with the decision stump, and AdaBoost with the decision tree, implemented in the Java programming language for compatibility with Android based devices. The configurations used were different for the different methods implemented. Firstly, the DNN method was implemented with an activation function named sigmoid, which is a function that has the sigmoid curve, widely used as an activation function for neural networks [39]. Several learning rates were previously studied, and it was verified that we obtained better results with a value equal to 0.1. For this method, the maximum number of training iterations was established as  $4 \times 10^6$ . The method was implemented without distance weighting, with three hidden layers, a seed value of six, and backpropagation. The Xavier function [40] was used as an initialization function, implementing  $L_2$  regularization [41]. Secondly, the IBk method was implemented with a batch size of 100, a  $k$  value of 1, and the linear nearest neighbour search algorithm [42]. Finally, in the last two methods implemented, the main difference was the weak classifier used in combination with the AdaBoost method as the decision stump classifier [43], for the first one, and the decision tree classifier [44], for the second one. Other differences were revealed, where the combination of the

AdaBoost method with the decision stump classifier was implemented with a maximum number of training iterations as 10, a seed value of 1, a batch size of 100, a weight threshold of 100, and without resampling. Thus, the combination of the AdaBoost method with the decision tree classifier was implemented with a seed value of 2, a batch size of 10, a number of maximum nodes equal to 4, and 200 as the number of trees.

Initially, we started with the identification of simple ADL, i.e., walking running, standing, going upstairs, and going downstairs, which was performed with the data acquired from the accelerometer, magnetometer, and gyroscope sensors. Secondly, the recognition of environments, i.e., bar, classroom, gym, library, street, hall, living room, kitchen, and bedroom, was performed with the data retrieved from the microphone. Finally, the recognition of activities without motion, i.e., driving, sleeping, and watching television, was performed with the data collected by the accelerometer, magnetometer, gyroscope, and GPS receiver with the inclusion of the environment recognised. Thus, the framework provided the recognition of eight ADL and nine environments.

For the implementation of the methods, the following technologies and frameworks were used:

- DNN: DeepLearning4j framework [45];
- IBk: Weka software [35];
- AdaBoost with the decision stump: Weka software [35];
- AdaBoost with the decision tree: Smile (Statistical Machine Intelligence and Learning Engine) framework [36].

### 3. Results

#### 3.1. Recognition of Simple ADL

The results of simple ADL recognition with the IBk method presented around 80% accuracy using the different combinations of motion and magnetic sensors, as presented in Table 2.

**Table 2.** ADL recognition using the Instance Based k-nearest neighbour (IBk) method implemented with Weka software.

Sensors	Correlation Coefficient	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error	Accuracy
Accelerometer	0.8335	0.261	0.817	21.8138%	57.7675%	73.9%
Accelerometer and Magnetometer	0.8771	0.2076	0.7011	17.2911%	49.5751%	79.23%
Accelerometer, Magnetometer, and Gyroscope	0.8781	0.2009	0.6991	16.733%	49.4287%	79.91%

AdaBoost is a binary classifier that uses a weak classifier to improve the recognition of different events. The implementation of this algorithm was performed with the identification of each ADL. The results of simple ADL identification with the AdaBoost with the decision stump method implemented with Weka software are presented in Table 3, verifying that all of the ADL were recognised with an accuracy between 25.61% (going downstairs recognised with the accelerometer and magnetometer sensors) and 98.44% (standing recognised with the accelerometer, magnetometer, and gyroscope sensors).

**Table 3.** Accuracies of ADL recognition using the AdaBoost with the decision stump method implemented with Weka software.

ADL	Accelerometer	Accelerometer and Magnetometer	Accelerometer, Magnetometer, and Gyroscope
Going downstairs	26.24%	25.61%	37.79%
Going upstairs	31.73%	32.64%	32.91%
Running	93.13%	93.00%	92.26%
Standing	96.35%	96.58%	98.44%
Walking	37.51%	51.23%	50.87%

In addition, Table 4 presents the clarification of the values obtained in Table 3, presenting the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. As this recognition was performed as binary recognition, i.e., the comparisons were performed by comparing the correct value with all records, we verified that the values of TP and TN were higher than others, proving the reliability of the method.

**Table 4.** Confusion matrix values of ADL recognition using the AdaBoost with the decision stump method implemented with Weka software (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative).

ADL	Accelerometer				Accelerometer and Magnetometer				Accelerometer, Magnetometer and Gyroscope			
	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP
Going downstairs	7469	1061	531	939	7467	1073	533	927	7606	1017	394	983
Going upstairs	7075	630	925	1370	7379	967	621	1033	7627	1498	373	502
Running	7919	81	81	1919	7914	82	86	1918	7917	97	83	1903
Standing	7938	26	62	1974	7933	33	67	1967	7977	23	23	1977
Walking	7472	552	528	1448	7629	632	371	1368	7609	546	391	1454

Moreover, the results on the recognition of simple ADL with AdaBoost with the decision tree method implemented with the Smile framework are presented in Table 5, verifying that all of the ADL presented an accuracy between 83.79% and 99.55% using the different combinations of motion and magnetic sensors.

Additionally, Table 6 presents the clarification of the values obtained in Table 5, presenting the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. As this recognition was performed as binary recognition, i.e., the comparisons were performed by comparing the correct value with all records, we verified that the sum of the values of TP and TN was 2000. This was the value of the number of samples equal to each activity, but the method reported a high number of FP.

Finally, the results previously obtained with the implementation of the recognition of simple ADL with the DNN method implemented with the Deeplearning4j framework are presented in Table 7, verifying that all of the ADL showed an accuracy between 66.70% and 99.35% using the different combinations of motion and magnetic sensors.

**Table 5.** Accuracies of ADL identification using AdaBoost with the decision tree implemented with the SMILE framework.

ADL	Accelerometer	Accelerometer and Magnetometer	Accelerometer, Magnetometer, and Gyroscope
Going downstairs	83.79%	84.21%	86.07%
Going upstairs	85.29%	84.70%	85.44%
Running	98.49%	98.47%	98.43%
Standing	99.04%	99.01%	99.55%
Walking	86.90%	89.53%	91.13%

**Table 6.** Confusion matrix values of ADL identification using AdaBoost with the decision tree implemented with the SMILE framework (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative).

ADL	Accelerometer				Accelerometer and Magnetometer				Accelerometer, Magnetometer, and Gyroscope			
	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
Going downstairs	1017	983	7362	638	972	1028	7449	551	974	1026	7633	367
Going upstairs	1086	914	7443	557	940	1060	7530	470	1083	917	7461	539
Running	1917	83	7932	68	1917	83	7930	70	1908	92	7935	65
Standing	1965	35	7939	61	1963	37	7938	62	1976	24	7979	21
Walking	1060	940	7620	380	1317	683	7636	364	1494	506	7619	381

**Table 7.** Accuracies of ADL identification using the DNN method.

ADL	Accelerometer	Accelerometer and Magnetometer	Accelerometer, Magnetometer, and Gyroscope
Going downstairs	66.70%	67.95%	77.25%
Going upstairs	84.45%	81.55%	82.40%
Running	95.45%	95.70%	95.85%
Standing	99.25%	99.20%	99.35%
Walking	86.10%	88.05%	90.09%

### 3.2. Recognition of Environments

The use of the IBk method for the recognition of environments using the microphone data reported an average accuracy of 41.43%, as presented in Table 8. The remaining results presented in Table 9 showed that the AdaBoost with the decision stump method implemented with Weka software had an accuracy between 10.36% and 91.78%. Next, the AdaBoost with the decision tree implemented with the SMILE framework reported an accuracy between 88.74% and 99.08%. Finally, the DNN method implemented with the Deeplearning4j framework presented an accuracy between 19.90% and 98.00%.

In addition, Table 10 presents the clarification of the values obtained in Table 9, presenting the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. As this recognition was performed as binary recognition, i.e., the comparisons were performed by comparing the correct value with all records, we verified that the values of TP were higher in the recognition of bar, library, hall, and street. However, in the remaining classes, the values of TN were correctly recognised.

**Table 8.** Recognition of environments using the IBk method implemented with Weka software.

<b>Sensors</b>	Sound
<b>Correlation coefficient</b>	0.8171
<b>Mean absolute error</b>	0.5857
<b>Root mean squared error</b>	1.5574
<b>Relative absolute error</b>	26.3488%
<b>Root relative squared error</b>	60.3156%
<b>Accuracy</b>	41.43%

**Table 9.** Accuracies of recognition of environments using the AdaBoost and DNN methods.

<b>Environments</b>	<b>AdaBoost with the Decision Stump</b>	<b>AdaBoost with the Decision Tree</b>	<b>DNN</b>
Bar	91.78%	99.08%	22.05%
Classroom	20.67%	88.74%	37.95%
Gym	10.36%	88.87%	87.85%
Hall	40.36%	92.38%	34.80%
Kitchen	16.11%	88.89%	51.35%
Library	34.01%	91.59%	19.90%
Street	38.38%	90.92%	25.35%
Bedroom	17.88%	88.88%	98.60%
Living room	18.82%	89.20%	33.50%

**Table 10.** Confusion matrix values of the recognition of environments using AdaBoost with the decision stump implemented with Weka software (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative).

<b>ADL</b>	<b>Sound</b>			
	<b>TN</b>	<b>FP</b>	<b>FN</b>	<b>TP</b>
Bar	15,961	146	39	1854
Library	15,791	1183	209	817
Hall	15,119	645	881	1355
Kitchen	16,000	1999	0	1
Bedroom	16,000	1999	0	1
Street	15,517	1180	483	820
Classroom	16,000	1999	0	1
Living room	16,000	1999	0	1
Gym	16,000	1999	0	1

Furthermore, Table 11 presents the clarification of the values obtained in Table 5, presenting the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. As this recognition was performed as binary recognition, i.e., the comparisons were performed comparing the correct value with all records, we verified that the values of TP were higher in the recognition of bar, library, hall, and street. However, in the remaining classes, the values of TN were also correctly recognised.



**Table 11.** Confusion matrix values of the recognition of environments using AdaBoost with the decision tree implemented with the SMILE framework (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative).

ADL	Sound			
	TP	FP	TN	FN
Bar	1917	83	15,918	82
Library	720	1280	15,767	233
Hall	1419	581	15,210	790
Kitchen	1	1999	16,000	0
Bedroom	14	1986	15,984	16
Street	787	1213	15,579	421
Classroom	148	1852	15,825	175
Living room	168	1832	15,888	112
Gym	1	1999	15,995	5

### 3.3. Recognition of Activities without Motion

Initially, we presented, in Table 12, the results on the recognition of activities without motion with the IBk method reporting an accuracy between 99.27% and 100% using the data acquired from the accelerometer, magnetometer, gyroscope, GPS receiver, and the environment previously identified.

**Table 12.** Accuracies of the recognition of activities without motion using the IBk method implemented with Weka software.

Sensors	Correlation Coefficient	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error	Accuracy
Accelerometer and environment	1	0	0	0	0	100%
Accelerometer, Magnetometer, and Environment	1	0	0	0	0	100%
Accelerometer, Magnetometer, Gyroscope, and Environment	1	0	0	0	0	100%
Accelerometer, Distance, and Environment	0.9969	0.0042	0.0645	0.6235%	7.903%	99.58%
Accelerometer, Magnetometer, Distance, and Environment	0.9964	0.0045	0.0695	0.6734%	8.5118%	99.55%
Accelerometer, Magnetometer, Gyroscope, Distance, and Environment	0.9943	0.0073	0.0876	1.0974%	10.7201%	99.27%

Furthermore, the results of the implementation of the recognition of activities without motion with the AdaBoost with the decision stump method implemented with Weka software are presented in Tables 13 and 14, verifying that the events were recognised with an accuracy between 98.32% and 100% using the data acquired from the accelerometer, magnetometer, gyroscope, GPS receiver, and the environment previously identified.

**Table 13.** Accuracies of the activities’ recognition without motion using the AdaBoost with the decision stump method implemented with Weka software for motion and magnetic sensors after the recognition of the environment.

	<b>Accelerometer and Environment</b>	<b>Accelerometer, Magnetometer, and Environment</b>	<b>Accelerometer, Magnetometer, Gyroscope, and Environment</b>
Watching television	100%	100%	100%
Sleeping	100%	100%	100%

**Table 14.** Accuracies of the activities’ recognition without motion using the AdaBoost with the decision stump method implemented with Weka software for motion, magnetic, and location sensors after the recognition of the environment

	<b>Accelerometer, Distance, and Environment</b>	<b>Accelerometer, Magnetometer, Distance, and Environment</b>	<b>Accelerometer, Magnetometer, Gyroscope, Distance, and Environment</b>
Watching television	98.58%	98.98%	98.98%
Driving	100%	100%	100%
Sleeping	98.32%	98.32%	98.32%

Additionally, Tables 15 and 16 present the clarification of the values obtained in Tables 13 and 14, presenting the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. As this recognition was performed as binary recognition, i.e., the comparisons were performed by comparing the correct value with all records, we verified that the values of TP and TN were higher than others, proving the reliability of the method.

**Table 15.** Confusion matrix values of the recognition of activities without motion using the AdaBoost with the decision stump method implemented with Weka software for motion and magnetic sensors after the recognition of the environment (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative).

ADL	<b>Accelerometer and Environment</b>				<b>Accelerometer, Magnetometer, and Environment</b>				<b>Accelerometer, Magnetometer, Gyroscope, and Environment</b>			
	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP
Watching television	2000	0	0	2000	2000	0	0	2000	2000	0	0	2000
Sleeping	2000	0	0	2000	2000	0	0	2000	2000	0	0	2000

**Table 16.** Confusion matrix values of the recognition of activities without motion using the AdaBoost with the decision stump method implemented with Weka software for motion, magnetic, and location sensors after the recognition of the environment (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative)

ADL	Accelerometer, Distance, and Environment				Accelerometer, Magnetometer, Distance, and Environment				Accelerometer, Magnetometer, Gyroscope, Distance, and Environment			
	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP
Watching television	3979	0	21	2000	3998	13	2	1987	3998	13	2	1987
Driving	4000	1	0	1999	4000	1	0	1999	4000	1	0	1999
Sleeping	3974	0	26	2000	3974	0	26	2000	3974	0	26	2000

Additionally, the results on the recognition of activities without motion with the AdaBoost with the decision tree implemented with the SMILE framework are presented in Tables 17 and 18, verifying that the events were recognised with an accuracy between 98.50% and 100% using the data acquired from the accelerometer, magnetometer, gyroscope, GPS receiver, and the environment previously identified.

**Table 17.** Accuracies of the activities' recognition without motion using the AdaBoost with the decision tree implemented with the SMILE framework for motion and magnetic sensors after the recognition of the environment.

	Accelerometer and Environment	Accelerometer, Magnetometer, and Environment	Accelerometer, Magnetometer, Gyroscope, and Environment
Watching television	100%	100%	100%
Sleeping	100%	100%	100%

**Table 18.** Accuracies of the activities' recognition without motion using the AdaBoost with the decision tree implemented with the SMILE framework for motion, magnetic, and location sensors after the recognition of the environment.

	Accelerometer, Distance, and Environment	Accelerometer, Magnetometer, Distance, and Environment	Accelerometer, Magnetometer, Gyroscope, Distance, and Environment
Watching television	99.67%	99.97%	99.97%
Driving	99.98%	99.98%	99.98%
Sleeping	99.52%	99.52%	99.50%

Tables 19 and 20 present the clarification of the values obtained in Tables 17 and 18, presenting the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. As this recognition was performed as binary recognition, i.e., the comparisons were performed comparing the correct value with all records, we verified that the values of TP and TN were higher than others, proving the reliability of the method.

**Table 19.** Confusion matrix values of the recognition of activities without motion using the AdaBoost with the decision tree implemented with the SMILE framework for motion and magnetic sensors after the recognition of the environment (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative).

ADL	Accelerometer and Environment				Accelerometer, Magnetometer, and Environment				Accelerometer, Magnetometer, Gyroscope, and Environment			
	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN
Watching television	2000	0	2000	0	2000	0	2000	0	2000	0	2000	0
Sleeping	2000	0	2000	0	2000	0	2000	0	2000	0	2000	0

**Table 20.** Confusion matrix values of the recognition of activities without motion using the AdaBoost with the decision tree implemented with the SMILE framework for motion, magnetic, and location sensors after the recognition of the environment (TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative).

ADL	Accelerometer, Distance, and Environment				Accelerometer, Magnetometer, Distance, and Environment				Accelerometer, Magnetometer, Gyroscope, Distance, and Environment			
	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN
Watching television	2000	0	3980	20	2000	0	3998	2	2000	0	3998	2
Driving	1999	1	4000	0	1999	1	4000	0	1999	1	4000	0
Sleeping	1998	2	3973	27	1998	2	3973	27	1998	2	3972	28

Finally, the results of the activity recognition without motion using the DNN method implemented with the DeepLearning4j framework are presented in Tables 21 and 22, verifying that the events were recognised with an accuracy between 79.55% and 98.50% using the data acquired from the accelerometer, magnetometer, gyroscope, GPS receiver, and the environment previously identified.

**Table 21.** Accuracies of the activities' recognition without motion using the DNN method for motion and magnetic sensors after the recognition of the environment.

	Accelerometer and Environment	Accelerometer, Magnetometer, and Environment	Accelerometer, Magnetometer, Gyroscope, and Environment
Watching television	94.05%	94.00%	94.15%
Sleeping	97.90%	97.85%	98.00%

Based on the results reported, Table 23 presents the average of the results obtained with the different algorithms implemented. As shown, the best results were achieved with the IBk method (99.68%) and AdaBoost with the decision tree as a weak classifier (94.05%).

The training stage was faster with IBk and AdaBoost with the decision tree than the DNN method previously implemented. These methods were less complicated to implement than the DNN method and were more efficient.

**Table 22.** Accuracies of the activities' recognition without motion using using the DNN method for motion, magnetic, and location sensors after the recognition of the environment.

	<b>Accelerometer, Distance, and Environment</b>	<b>Accelerometer, Magnetometer, Distance, and Environment</b>	<b>Accelerometer, Magnetometer, Gyroscope, Distance, and Environment</b>
Watching television	94.15%	94.25%	94.35%
Driving	80.65%	79.55%	84.15%
Sleeping	98.50%	98.30%	98.15%

**Table 23.** Average of the accuracy of each implemented method.

<b>Stages</b>	<b>DNN</b>	<b>IBk</b>	<b>AdaBoost with the Decision Stump</b>	<b>AdaBoost with the Decision Tree</b>
Stage 1	87.29%	77.68%	59.75%	91.33%
Stage 2	45.71%	41.43%	32.04%	90.95%
Stage 3	99.87%	99.73%	92.83%	99.87%
Overall	77.62%	72.95%	61.54%	94.05%

Based on the limitations of mobile devices, these methods should be implemented in the ADL and environment recognition framework to improve the results provided to the user. The results showed that the recognition of ADL and its environments was possible with the implementation of the AdaBoost, IBk, and DNN methods. It allows opportunities to create a personal digital life coach and monitor the different lifestyles. It is important for all people, because mobile devices are widely used. They exploit the possibilities to improve the quality of life.

#### 4. Discussion and Conclusions

The implementations of DNN, IBk, AdaBoost with the decision stump, and AdaBoost with the decision tree were performed with success with the dataset previously acquired, which was based on the data received from the accelerometer, magnetometer, gyroscope, GPS receiver, and microphone. The framework was composed of data acquisition, data processing, data cleaning, feature extraction, data fusion, and data classification, to recognise eight ADL and nine environments.

In general, the overall accuracies of the methods depended on the number of sensors and resources available during data acquisition. The framework should be a function of the number of sensors available in mobile devices. The methods with an accuracy higher than 90% were the IBk method and AdaBoost with the decision tree as the weak classifier.

The AdaBoost and IBk methods reported the best results because these methods were not susceptible to overfitting in comparison with the DNN method. Notably, one of the reasons for this conclusion was the use of a weak classifier by AdaBoost that handled the discrimination of some results.

According to the previously proposed structure of a framework for the recognition of ADL and environments [2,17–25], the main focus of this study was related to the data classification module, taking into account the implementations of the other modules performed in previous studies. Previously, the DNN method was implemented, and it reported reliable results. Still, for the recognition of the environments with acoustic data, the results obtained were below the expectations, because it took many resources from the processing unit. For the validation of the different implemented methods, we performed cross-validation with 10 folds.

Following the tests of the different methods for the recognition of simple ADL, the best results were achieved with AdaBoost with the decision tree implemented with the SMILE framework, reporting an overall accuracy of 91.33% with all combinations of sensors. Still, there was a high number of FP.

In the case of the recognition of environments, the best method was also AdaBoost with the decision tree implemented with the SMILE framework, reporting an overall accuracy of 99.87%. Still, it did not recognise correctly two environments. However, the AdaBoost with the decision stump method implemented with Weka software did not recognise five environments correctly, reporting an overall accuracy of 32.04%. Finally, in the recognition of activities without motion, the results obtained with AdaBoost with the decision tree implemented with the SMILE framework were the same as the results obtained with the DNN method (99.87%).

As future work, the methods should be implemented during the development of the framework for the identification of ADL and its environments, adapting the approach to all the sensors available on mobile devices.

**Author Contributions:** Conceptualization, methodology, software, validation, formal analysis, investigation, writing, original draft preparation, and writing, review and editing: J.M.F., I.M.P., G.M., N.M.G., E.Z., P.L., F.F.-R., S.S., and L.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project **UIDB/EEA/50008/2020** (*Este trabalho é financiado pela FCT/MCTES através de fundos nacionais e quando aplicável cofinanciado por fundos comunitários no âmbito do projeto UIDB/EEA/50008/2020*).

**Acknowledgments:** This work is funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project **UIDB/EEA/50008/2020** (*Este trabalho é financiado pela FCT/MCTES através de fundos nacionais e quando aplicável cofinanciado por fundos comunitários no âmbito do projeto UIDB/EEA/50008/2020*). This article is based on work from COST Action IC1303 - AAPELE - Architectures, Algorithms and Protocols for Enhanced Living Environments, and COST Action CA16226 - SHELD-ON- Indoor living space improvement: Smart Habitat for the Elderly, supported by COST (European Cooperation in Science and Technology). More information at [www.cost.eu](http://www.cost.eu).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Mobile Marketing Statistics Compilation | Smart Insights. Smart Insights, 2019. Available online: <https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/> (accessed on 11 November 2019).
2. Pires, I.; Garcia, N.; Pombo, N.; Flórez-Revuelta, F. From Data Acquisition to Data Fusion: A Comprehensive Review and a Roadmap for the Identification of Activities of Daily Living Using Mobile Devices. *Sensors* **2016**, *16*, 184. [[CrossRef](#)] [[PubMed](#)]
3. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F.; Rodríguez, N.D. Validation Techniques for Sensor Data in Mobile Health Applications. *J. Sens.* **2016**, *2016*, 1687–725. [[CrossRef](#)]
4. Shuib, L.; Shamshirb, S.; Ismail, M.H. A review of mobile pervasive learning: Applications and issues. *Comput. Hum. Behav.* **2015**, *46*, 239–244. [[CrossRef](#)]
5. Garcia, N.M.; Rodrigues, J.J.P. (Eds.). *Ambient Assisted Living*; CRC Press: Boca Raton, FL, USA, 2015.
6. Garcia, N.M. Roadmap to the Design of a Personal Digital Life Coach. In *International Conference on ICT Innovations*; Springer: Cham, Switzerland, 2015; pp. 21–27.
7. Sousa, P.S.; Sabugueiro, D.; Felizardo, V.; Couto, R.; Pires, I.; Garcia, N.M. mHealth sensors and applications for personal aid. In *Mobile Health*; Springer: Cham, Switzerland, 2015; pp. 265–281.
8. Dobre, C.; Mavromoustakis, C.X.; Garcia, N.M.; Mastorakis, G.; Goleva, R.I. Introduction to the AAL and ELE Systems. In *Ambient Assisted Living and Enhanced Living Environments*; Butterworth-Heinemann: Oxford, UK, 2017; pp. 1–16.
9. Felizardo, V.; Sousa, P.; Sabugueiro, D.; Alexre, C.; Couto, R.; Garcia, N.; Pires, I. E-Health: Current status and future trends. In *Handbook of Research on Democratic Strategies and Citizen-Centered E-Government Services*; IGI Global: Hershey, PA, USA, 2015; pp. 302–326.
10. Goleva, R.I.; Garcia, N.M.; Mavromoustakis, C.X.; Dobre, C.; Mastorakis, G.; Stainov, R.; Trajkovik, V. AAL and ELE Platform Architecture. In *Ambient Assisted Living and Enhanced Living Environments*; Butterworth-Heinemann: Oxford, UK, 2017; pp. 171–209.
11. Banos, O.; Damas, M.; Pomares, H.; Rojas, I. On the use of sensor fusion to reduce the impact of rotational and additive noise in human activity recognition. *Sensors* **2012**, *12*, 8039–8054. [[CrossRef](#)]

12. Akhoundi, M.A.A.; Valavi, E. Multi-Sensor Fuzzy Data Fusion Using Sensors with Different Characteristics. *arXiv* **2010**, arXiv:1010.6096.
13. Paul, P.; George, T. An Effective Approach for Human Activity Recognition on Smartphone. In Proceedings of the 2015 IEEE International Conference on Engineering and Technology (Icetech), Coimbatore, India, 25 January 2015; pp. 45–47. [[CrossRef](#)]
14. Hsu, Y.-W.; Chen, K.-H.; Yang, J.-J.; Jaw, F.-S. Smartphone based fall detection algorithm using feature extraction. In Proceedings of the 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Datong, China, 15 October 2016; pp. 1535–1540.
15. Dernbach, S.; Das, B.; Krishnan, N.C.; Thomas, B.L.; Cook, D.J. Simple and Complex Activity Recognition through Smart Phones. In Proceedings of the 2012 8th International Conference on Intelligent Environments (IE), Guanajuato, Mexico, 14 January 2012; pp. 214–221.
16. Shen, C.; Chen, Y.F.; Yang, G.S. On Motion-Sensor Behavior Analysis for Human-Activity Recognition via Smartphones. In Proceedings of the 2016 IEEE International Conference on Identity, Security and Behavior Analysis (Isba), Sendai, Japan, 22 January 2016; pp. 1–6.
17. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F. Identification of Activities of Daily Living Using Sensors Available in off-the-shelf Mobile Devices: Research and Hypothesis. In *International Symposium on Ambient Intelligence*; Springer: Cham, Switzerland, 2016; pp. 121–130.
18. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F.; Spinsante, S. Pattern recognition techniques for the identification of Activities of Daily Living using mobile device accelerometer. *arXiv* **2017**, arXiv:1711.00096.
19. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F.; Spinsante, S.; Goleva, R.; Zdravevski, E. Recognition of activities of daily living based on environmental analyses using audio fingerprinting techniques: A systematic review. *Sensors* **2018**, *18*, 160. [[CrossRef](#)]
20. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F.; Spinsante, S. Approach for the development of a framework for the identification of activities of daily living using sensors in mobile devices. *Sensors* **2018**, *18*, 640. [[CrossRef](#)]
21. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F.; Spinsante, S.; Teixeira, M.C. Identification of activities of daily living through data fusion on motion and magnetic sensors embedded on mobile devices. In *Pervasive and Mobile Computing*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 47, pp. 78–93.
22. Pires, I.M.; Teixeira, M.C.; Pombo, N.; Garcia, N.M.; Flórez-Revuelta, F.; Spinsante, S.; Goleva, R.; Zdravevski, E. Android Library for Recognition of Activities of Daily Living: Implementation Considerations, Challenges, and Solutions. *Open Bioinform. J.* **2018**. [[CrossRef](#)]
23. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F. Framework for the Recognition of Activities of Daily Living and their Environments in the Development of a Personal Digital Life Coach. *DATA* **2018**. [[CrossRef](#)]
24. Pires, I.M.S. Multi-Sensor Data Fusion in Mobile Devices for the Identification of Activities of Daily Living. Ph.D. Thesis, Universidade da Beira Interior, Covilhã, Portugal, November 2018.
25. Pires, I.M.; Marques, G.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F.; Spinsante, S.; Teixeira, M.C.; Zdravevski, E. Recognition of Activities of Daily Living and Environments Using Acoustic Sensors Embedded on Mobile Devices. *Electronics* **2019**, *8*, 1499. [[CrossRef](#)]
26. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)] [[PubMed](#)]
27. Costarelli, D.; Vinti, G. Pointwise and uniform approximation by multivariate neural network operators of the max-product type. *Neural Netw.* **2016**, *81*, 81–90. [[CrossRef](#)] [[PubMed](#)]
28. Gripenberg, G. Approximation by neural networks with a bounded number of nodes at each level. *J. Approx. Theory* **2003**, *122*, 260–266. [[CrossRef](#)]
29. Pires, I.M.; Garcia, N.M.; Pombo, N.; Flórez-Revuelta, F. Limitations of the Use of Mobile Devices and Smart Environments for the Monitoring of Ageing People. In Proceedings of the 4th International Conference on Information and Communication Technologies for Ageing Well and e-Health, Madeira, Portugal, 22–23 March 2018; pp. 269–275.
30. Pires, I.; Felizardo, V.; Pombo, N.; Garcia, N.M. Limitations of energy expenditure calculation based on a mobile phone accelerometer. In Proceedings of the 2017 International Conference on High Performance Computing & Simulation (HPCS), Genoa, Italy, 17–21 July 2017.

31. August 2017—Multi-Sensor Data Fusion in Mobile Devices for the Identification of Activities of Daily Living. Available online: [https://github.com/impire/August\\_2017-Multi-sensor\\_data\\_fusion\\_in\\_mobile\\_devices\\_for\\_the\\_identification\\_of\\_activities\\_of\\_dail](https://github.com/impire/August_2017-Multi-sensor_data_fusion_in_mobile_devices_for_the_identification_of_activities_of_dail) (accessed on 20 February 2019).
32. Yoav, F.; Schapire, R.E. A Decision-Theoretic Generalisation of on-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1995**, *55*, 119.
33. Hastie, T.; Rosset, S.; Zhu, J.; Zou, H. Multi-class AdaBoost, 2009. *Stat. Interface* **2008**, *2*, 349–360. [[CrossRef](#)]
34. Pollettini, J.T.; Panico, S.R.; Daneluzzi, J.C.; Tinós, R.; Baranauskas, J.A.; Macedo, A.A. Using machine learning classifiers to assist healthcare-related decisions: Classification of electronic patient records. *J. Med. Syst.* **2012**, *36*, 3861–3874. [[CrossRef](#)]
35. Frank, E.; Hall, M.; Reutemann, P.; Trigg, L. Weka 3—Data Mining with Open Source Machine Learning Software in Java, 2019. Available online: <https://www.cs.waikato.ac.nz/ml/Weka/index.html> (accessed on 10 November 2019).
36. Github, Smile—Statistical Machine Intelligence and Learning Engine, 2019. Available online: <http://haifengl.github.io/smile/> (accessed on 10 November 2019).
37. Graizer, V. Effect of low-pass filtering and re-sampling on spectral and peak ground acceleration in strong-motion records. In Proceedings of the 15th World Conference of Earthquake Engineering, Lisbon, Portugal, 28 September 2012; pp. 24–28.
38. Rader, C.; Brenner, N. A new principle for fast Fourier transformation. *IEEE Trans. Acoust. Speech Signal Process.* **1976**, *24*, 264–266. [[CrossRef](#)]
39. Karlik, B.; Olgac, A.V. Performance analysis of various activation functions in generalized MLP architectures of neural networks. *Int. J. Artif. Intell. Expert Syst.* **2011**, *1*, 111–122.
40. Kumar, S.K. On weight initialization in deep neural networks. *arXiv* **2017**, arXiv:1704.08863, 2017.
41. Van Laarhoven, T. L2 regularization versus batch and weight normalization. *arXiv* **2017**, arXiv:1706.05350.
42. Nene, S.A.; Nayar, S.K. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 989–1003. [[CrossRef](#)]
43. Kawaguchi, S.; Nishii, R. Hyperspectral image classification by bootstrap AdaBoost with random decision stumps. *IEEE Trans. Geosci. Remote. Sens.* **2007**, *45*, 3845–3851. [[CrossRef](#)]
44. Safavian, S.R.; Lgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [[CrossRef](#)]
45. Nicholson, A.C. DeepLearning4j: Open-source, Distributed Deep Learning for the JVM, 2 September 2017. Available online: <https://deeplearning4j.org/> (accessed on 10 November 2019).



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).